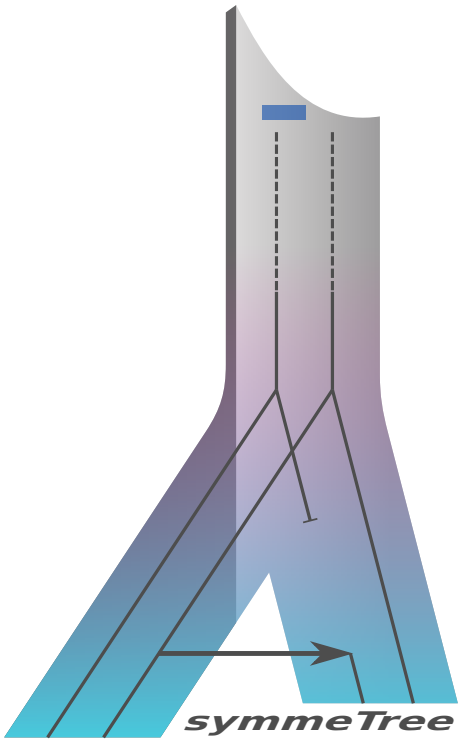

AsymmeTree User Manual

David Schaller

sdavid@bioinf.uni-leipzig.de

Contents

1 Introduction

AsymmeTree is an open-source Python library for the simulation and analysis of phylogenetic scenarios. It includes a simulator for species and gene trees with asymmetric evolution rates, tools for the inference and analysis of phylogenetic best matches [27, 28] (resp. best hits) from known gene trees or evolutionary distances. Moreover, it includes an algorithm to compute supertrees [20] and a method to estimate rooted species trees from an ensemble of orthology/paralogy relations [37].

The library, and especially the simulator, was primarily designed to be able to validate mathematical concepts and test inference methods for various steps on the way to more realistically available data, i.e., dated gene trees, additive distances of gene sets, noisy distances and finally sequences. Both nucleotide and amino acid sequence simulation with or without indels are supported. In both cases, several substitution models are available. Alternatively, third-party software such as `Pyvolve` [73] can easily be incorporated into a simulation pipeline based on AsymmeTree.

The software is hosted on GitHub and also available via The Python Package Index (PyPI). Please feel free to report bugs or make suggestions for improvement in the Issues section of the GitHub repository.

If you use AsymmeTree in your project or code from it, please cite:

> Peter F. Stadler, Manuela Geiß, David Schaller, Alitzel López Sánchez, Marcos González Laffitte, Dulce I. Valdivia, Marc Hellmuth, Maribel Hernández Rosales (2020). **From pairs of most similar sequences to phylogenetic best matches.** *Submitted to Algorithms for Molecular Biology.*

This document is split into two parts. Chapter 2 is a manual that describes the installation and usage of the package, whereas Chapter 3 contains detailed descriptions of the motivation and implementation of the simulator's main components.

2 Manual

2.1 Installation

AsymmeTree requires Python 3.5 or higher. Python 2 is not supported.

2.1.1 Easy Installation with pip

The `asymmetree` package is available on The Python Package Index (PyPI):

```
pip install asymmetree
```

For details about how to install Python packages see here.

2.1.2 Installation with the setup file

Alternatively, you can download or clone the repo, go to the root folder of package and install it using the command:

```
python setup.py install
```

2.1.3 Dependencies

AssymmeTree has several dependencies (which are installed automatically when using `pip` or the `setup.py`):

- NetworkX
- SciPy and NumPy
- Matplotlib

To use the tree reconstruction method for best match inference and the C++ implementation of the quartet method [74], resp., the following software must be installed (I recommend that you compile these tools on your machine, place the binaries into a persistent location and add this location to your PATH environment variable):

- RapidNJ [72]
- qinfer

2.2 Overview

Table 2.1 is an overview over the subpackages and modules that may be relevant for users.

Packages and Modules	Description
`simulator`	
`TreeSimulator`	Simulator for dated species trees and dated gene trees, construction of the observable gene tree.
`TreeImbalancer`	Simulation of evolution rate asymmetries, autocorrelation between ancestors and descendants as well as correlation between genes in the same species.
`Scenario`	Wrapper class for species and gene tree scenarios, computation of the (R)BMG as well as event counts and some statistics.
`NoisyMatrix`	Generation of a noisy matrix (*random perturbation* or *wrong topology noise*).
`tools`	
`Tree`	Includes the basic class `Tree`, provides functions for traversals, Newick parser, etc.
`PhyloTree`	Includes the class `PhyloTree` for phylogenetic trees (inherits from `Tree`), provides a Newick parser, etc.
`BuildST`	Includes the class `BuildST` that computes a supertree from a given list of tress (with overlapping labels) [20].
`best_match_infer`	
`TrueBMG`	Computation of the true (R)BMG from a gene tree as well as the true orthology relation.
`ExtBestHits`	Implementation of the *Extended Best Hits* method, optionally uses `qinfer`.
`TreeReconstruction`	Reconstruction of the gene tree with `RapidNJ` [72] and midpoint rooting.
`Quartets`	Implementation of *Quartet* approach with two different methods for outgroup selection, optionally uses `qinfer`.
`LRTConstructor`	Construction of a least resolved tree (LRT) from a BMG via *informative triples*, optionally uses minimal edge cuts.
`cograph`	
`Cograph`	Includes the classes `Cotree` and `CotreeNode` as well as a generator for random cotrees/cographs.
`CographEditor`	Implements a heuristic for cograph editing [17].
`LinearCographDetector`	Implements an $\mathcal{O}(n+m)$ algorithm for cograph detection [16].
`paraphylo`	
`SpeciesTreeFromParalogs`	Species tree reconstruction from orthology/paralogy relations. Heuristic version of `ParaPhylo` [37].
`SpeciesTreeFromPO`	Species tree reconstruction from `ProteinOrtho` [54] output.
`visualize`	
`GeneTreeVis`	Visualization of a (simulated) gene tree (of type `PhyloTree`)

Table 2.1: Overview over the subpackages and modules.

2.2.1 Tree Data Structures

The two classes `Tree` and `PhyloTree` (inherits from `Tree`) implement tree data structures which are essential for most of the modules in the package. The latter contains converters and parsers for the Newick format and a NetworkX graph format.

The vertices of a `PhyloTree` instance are of type `PhyloTreeNode` and contain the following attributes:

`ID`	vertex ID (`int`)
`label`	label (`str`), in gene trees: `"S"` for speciation, `"D"` for duplication, `"H"` for horizontal gene transfer, `"*"` for loss
`color`	only gene trees; species in which the gene resides, i.e., ID of some vertex in a species tree, can be of type `tuple` (edge) for inner and loss vertices
`tstamp`	time stamp of the event (`double`)
`dist`	evolutionary distance or divergence time from the parent vertex (`double`)
`tranferred`	only gene trees; indicates whether the edge from the parent is the transfer edge from an HGT event; `1` if yes and `0` otherwise

Both species and gene trees can be converted into Newick format using the function `to_newick()` of the `PhyloTree` class. In case of a gene tree, the color is represented in brackets, e.g.

```
>>> "(3<1>:0.534,2<2>:0.762)S<0>:0.273"
```

To suppress this, use `to_newick(color=False)`. Likewise, to suppress the distances, you can use `to_newick(distance=False)`. The function `PhyloTree.parse_newick()` can handle this customized format as well as the standard Newick format.

If you intend to serialize species or gene trees, I recommend converting them into NetworkX graphs before applying Python's serialization library `pickle`. Note that the information about the ID of the root should be saved too:

```
import pickle

# tree is of type PhyloTree
tree_nx, root_id = tree.to_nx()

pickle.dump( (tree_nx, root_id), open("tree.pickle", "wb") )
```

To load a tree that was serialized this way use:

```
import pickle
from asymmetree.tools.PhyloTree import PhyloTree

tree_nx, root_id = pickle.load( open("tree.pickle", "rb") )
tree = PhyloTree.parse_nx(tree_nx, root_id)
```

2.2.2 Simulator for Species and Gene Trees

The subpackage `asymmetree.simulator` contains modules for the simulation and manipulation of species trees and gene trees.

Species trees

The function `build_species_tree(N)` simulates a dated species tree with `N` leaves (i.e. recent species) using the 'innovation model' described by Keller-Schmidt and Klemm [47]. The following keyword parameters (with their default value) are available:

`planted=True`	add a planted root that has the canonical root as its single neighbor, this way duplication (and loss) events can occur before the first speciation event in a subsequent gene tree simulation
`model="innovations"`	model for the species tree simulation, currently only the 'innovation model' is available
`non_binary=0.0`	probability that an inner edge is contracted, results in a non-binary tree

The time stamps of all vertices are normalized such that the root has time stamp `1.0` and all leaves have time stamp `0.0` (see Section 3.2.1).

Example usage:

```
import asymmetree.simulator.TreeSimulator as ts

S = ts.simulate_species_tree(10, planted=True, non_binary=0.2)
print(S.to_newick())
```

Gene trees

Dated gene trees are simulated along a given species tree `S` with a variant of the Gillespie algorithm [31]. To this end, an instance of the class `GeneTreeSimulator` must be initialized with a species tree of type `PhyloTree`. The following parameters are available (keyword arguments are indicated by `=default`):

`DLH_rates`	a `tuple` of three `floats`, rates for duplication, loss and HGT events in the Gillespie algorithm
`dupl_polytomy=0.0`	allows non-binary duplication events by specifying the lambda parameter for a poisson distribution (copy number = drawn number + 2)

At the moment, loss events in a branch are suppressed whenever this branch is the last survivor in its species branch (by setting the loss rate in the branch to zero). The behaviour is intended to be made optional in future releases.

Example usage:

```
import asymmetree.simulator.TreeSimulator as ts

# S is a species tree of type PhyloTree
TGT_simulator = ts.GeneTreeSimulator(S)
TGT = TGT_simulator.simulate(DLH_rates)
```

The function `observable_tree(tree)` returns the observable part of a gene tree, i.e., it copies the tree, removes all branches that lead to loss events only and suppresses all inner nodes with only one child. It also removes the planted root. Example usage:

```
# observable gene tree
OGT = ts.observable_tree(TGT)
```

Gene tree imbalancing

The module `TreeImbalancer` contains a function to model realistic (asymmetric) evolution rates for a given gene tree (see Section 3.2.3). Moreover, correlation in the evolution rate between genes of the same (and closely related) species is introduced (autocorrelation). The function `imbalance_tree(T, S)` takes a gene tree `T` and the **corresponding** species tree `S` as input and manipulated the branch length of the species tree. The following keyword parameters (with their default values) are available:

`baseline_rate=1.0`	starting value for the substitution rate (per time unit) and expected value for conserved genes
`autocorrelation_variance=0.0`	variance factor for a lognormal distribution that controls autocorrelation between genes of the same (and closely related) species, the higher the lower the autocorrelation
`gamma_param=(0.5, 1.0, 2.2)`	parameter the for Gamma distribution (`a, loc, scale`) from which rate factors for divergent are drawn, the default values are chosen to fit observed asymmetries between paralogs in yeast data [14]
`weights=(1, 1, 1)`	weights for choice between conservation, subfunctionalization and neofunctionalization after a duplication event
`inplace=True`	manipulate edge lengths (`dist`) of the gene tree in-place, otherwise copy the tree

In is recommended to apply the imbalancing to the true gene tree that still contains loss events. Example usage:

```
import asymmetree.simulator.TreeSimulator as ts
import asymmetree.simulator.TreeImbalancer as tm

S = ts.simulate_species_tree(10)

# true gene tree (with losses)
TGT_simulator = ts.GeneTreeSimulator(S)
TGT = TGT_simulator.simulate( (0.5, 0.5, 0.5) )    # event rates D, L, H

# imbalancing
TGT = tm.imbalance_tree(TGT, S, baseline_rate=1,
                        autocorrelation_variance=0.2,
                        gamma_param=(0.5, 1.0, 2.2),
                        weights=(1, 1, 1))

# observable gene tree
OGT = ts.observable_tree(TGT)
```

Distance matrix and noise simulation

The additive distance from an **observable** gene tree can be computed using the function `distance_matrix()` of a `PhyloTree` instance. It returns a tuple containing a list of leaves

in the tree (specifying the indexing) and the distance matrix as a 2-dimensional `numpy` array.

```
# T is an observable gene tree
leaves, D = T.distance_matrix()
leaf_index_dict = {leaf: i for i, leaf in leaves}
```

In the next step, noise can be introduced into a distance matrix using the `NoisyMatrix` module. Random noise can be simulated with the function `noisy_matrix(orig_matrix, sd)`. The following parameters are available (keyword arguments are indicated by their default value):

`orig_matrix`	original matrix to be disturbed
`sd`	standard deviation of a normal distribution with mean 1 from which noise factors are drawn
`metric_repair="reject"`	method to ensure that the resulting distance matrix is still a metric, available are `"reject"`, `"DOMR"` and `"general"` (see Section 3.2.4)

Alternatively, the function `convex_linear_comb(D1, D2)` can be used to simulate systematically biased noise by computing a linear convex combination with a disturbance matrix. The function thus takes two distance matrices (`numpy` arrays) not necessarily of the same size as input and disturbs them with one another. The contribution of the respective disturbance matrix is controlled by the keyword parameter `alpha` (default is `0.5`). If the keyword parameter `first_only` is specified as `True`, only the first disturbed matrix is returned. Otherwise both are returned in a tuple.

2.2.3 Best Match Inference

Phylogenetic best matches of a gene x of species X are defined as those genes y of another species $Y \neq X$ that share the lowest common ancestor with x in the gene tree among all genes in that species [27, 29, 28], see also Section 3.1.4. In contrast, two genes are orthologs if their last common ancestor was a speciation event. Orthology and reciprocal best matches are closely related [28].

The subpackage `asymmetree.best_matches` contains functions to compute both relations from a given gene tree or to estimate them from distance data on a set of genes [74].

If the true (observable) gene tree is known (as e.g. the case in simulations), best matches and orthologs can be computed using the module `TrueBMG`. The functions `best_match_graphs(tree)` and `true_orthology_graph(tree)` return the respective graph representation as NetworkX (di)graphs:

```
from asymmetree.best_matches.TrueBMG import true_orthology_graph,
                                            best_match_graphs

# T is an observable gene tree
orthology_graph = true_orthology_graph(T)
BMG, RBMG = best_match_graphs(T)     # Best Match Graph and Reciprocal Best
                                       Match Graph as a tuple
```

If only distance data is available, best matches have to be estimated. AsymmeTree currently implements three different methods that are described by Stadler et al. [74]:

- Extended Best Hits (module `ExtBestHits`)
- Neighborjoining and midpoint rooting (module `TreeReconstruction`, requires the installation and accessibility of `RapidNJ` [72])
- Quartet method (module `Quartets`, Python implementation and wrapper for the C++ tool `qinfer`)

Please see the file `examples/best_match_infer.py` in the GitHub repo for an example usage of these modules following the simulation of gene tree scenarios.

2.2.4 Supertree Computation

The module `BuildST` contains an implementation of the BuildST algorithm described by Deng and Fernández-Baca [20] to compute a supertree from a given list of tree based on the leaf labels. The algorithm uses the dynamic graph data structure described by Henzinger and King [39] and Holm et al. [42]. The latter can also be used separately (see module `asymmetree.tools.hdtgraph.DynamicGraph`)

The class `BuildST` is initialized with a list of trees that are of type `Tree` (thus also `PhyloTree` is allowed). The method `run()` then return a supertree if the trees in the list are compatible *and* they overlap in their sets of leaf labels. More precisely, the graph on the set of input trees in which two trees are connected by an edge if and only if they have at least one leaf label in common must be connected.

Example usage:

```
from asymmetree.tools.BuildST import BuildST

# tree_list is a list of Tree instances
st_builder = BuildST(tree_list)
supertree = st_builder.run()

if supertree:
  print(supertree.to_newick())
else:
  print("Could not build a supertree!")
```

2.2.5 Cograph Editing and ParaPhylo

The subpackages `asymmetree.cograph` and `asymmetree.proteinortho` contain heuristics for cograph editing and a method to compute rooted species tree from orthology/paralogy relations. The latter is a reimplementation of ParaPhylo [37] which uses heuristics for the NP-hard steps instead of exact ILP solutions. For cograph editing, the $\mathcal{O}(n^2)$ algorithm (where n is the number of vertices in a connected graph) by Crespelle [17] is applied. For the Maximum Consistent Triple Set problem, tree different heuristics are available:

- `BPMF`: Best-Pair-Merge-First [78] (modified for weighted triples)
- `MINCUT`: Aho's `BUILD` with weighted MinCut [1, 13]
- `GREEDY`: a greedy approch based on Aho's `BUILD`

The class `TreeReconstructor` in the module `SpeciesTreeFromParalogs` computes a species tree after it is provides with one or more NetworkX graphs that represent (estimated) orthology relations. To this end, the nodes in these graph must have a `"color"` attribute, since these will be the leaf labels in the reconstructed species tree. Example usage:

```
from asymmetree.paraphylo.SpeciesTreeFromParalogs import TreeReconstructor

tree_reconstr = TreeReconstructor()

# ortho_relations is a list of orthology relations
for graph in ortho_relations:
    tree_reconstr.add_ortho_graph(graph)

# finally the tree can be computed
S_estimate = tr.build_species_tree(mode="BPMF")
print(S_estimate.to_newick())
```

The module `SpeciesTreeFromPO` contains functions to estimate a species tree from a `ProteinOrtho` output file. For example, the function `reconstruct_from_PO(filename, triple_mode="BPMF")` takes the filename to the `ProteinOrtho` output file and optionally the triple heuristic as input, and returns a tuple consisting of the estimated species tree (`PhyloTree`) and a Newick representation (`str`) containing support values for the inner nodes.

3 Documentation of the Gene Family Simulator

A first version of this simulator for species trees and gene families was part of my Master's thesis [69] and has been used by Stadler et al. [74]. Parts of the following sections including some images have been reused from my thesis.

3.1 Mathematical Preliminaries

3.1.1 Graph and Tree Notation

In this section, a very short introduction into the field of graph theory is given with a focus on the concepts that will be relevant in the subsequent sections. For standard textbooks see e.g. [22] and Semple and Steel [70] from which definitions in the following were taken. The notation concerning trees follows the one used by Geiß et al. [27].

Definition 1 ((Di-)Graph)**.** *A* graph *is a pair $G = (V, E)$ of sets such that $E \subseteq [V]^2$, i.e., the elements of E are 2-element subsets of V. The elements in V are called vertices or nodes and the elements in E are called edges. Here, an (undirected) edge e between vertices x and y is written as xy.*

A directed graph *(or* digraph*) is s graph that assigns to every edge an initial and terminal vertex. In this case, an edge e is written as (x, y) if x is the initial, and y is the terminal vertex of edge e.*

As usual, it is often written $V(G) := V$ and $E(G) := E$ for the sets of vertices and edges of a graph $G = (V, E)$ in order to express affiliation. Two vertices x and y are *adjacent* if $xy \in E(G)$ [or $(x, y) \in E(G)$, respectively]. The *degree* of a vertex x is the number of adjacent vertices and denoted by $\mathsf{deg}(x)$.

The notation $G[x_1, ..., x_k]$ refers to an induced subgraph G' of a graph G on the set $V(G') = \{x_1, ..., x_k\} \subseteq V(G)$. Induced subgraphs are defined in the usual sense, i.e., $xy \in E(G')$ [or $(x, y) \in E(G')$ if G is directed] if and only if $xy \in E(G)$ [$(x, y) \in E(G)$] and $x, y \in V(G')$.

An undirected graph $G = (V, E)$ is called *connected* if any two vertices $x, y \in V(G)$ are linked by a path in G, i.e., there is a path $x = x_1 - x_2 - ... - x_k = y$ such that $x_i x_{i+1} \in E(G)$ for all $1 \leq i < k$. A maximal connected subgraph of G is called a *connected component* of G. In case of a directed graph $\vec{G}$, a subgraph $\vec{G}^*$ of $\vec{G}$ is called a *strongly connected component* if for any two vertices x and y there is directed path $x = x_1 \to x_2 \to ... \to x_k = y$ such that $(x_i, x_{i+1}) \in E(\vec{G})$ for all $1 \leq i < k$ and vice versa. Furthermore, $\vec{G}^*$ must be maximal in that sense.

Given a non-empty set of colors $\mathcal{C}$, a *proper vertex coloring* of a (di-)graph G is a map $\sigma \colon V(T) \to \mathcal{C}$ such that $xy \in E(G) \implies \sigma(x) \neq \sigma(y)$. In other words, two adjacent vertices must not have the same color.

Definition 2 (Rooted tree)**.** *A graph $T = (V, E)$ is a* tree *if it is connected and acyclic. A* rooted tree *is a tree T with a special node ρ called the* root*. As a consequence of this, a partial order $\preceq_T$ on the vertex set $V(T)$ can be defined where $v \preceq_T u$ if u lies on the (unique) path from ρ to v.*

A vertex u is called an *ancestor* of v in T if $v \preceq_T u$, whereas v is a descendant of u in this case. If furthermore $uv \in T$, then u is the parent of v, and v is a child of u. The set of all children of a vertex u is denoted by $\mathsf{child}(u)$. Likewise, $\mathsf{par}(v)$ refers to the unique parent of a vertex v. It is set $\mathsf{par}(v) = \emptyset$ if v has no parent, i.e., v is the root.

The set of all outer vertices $v \in V(T)$ (i.e., for which $\deg(v) = 1$) is called the *leaf set* of T and denoted by $L(T)$. Hence, the leafs $L(T)$ are the minima and ρ is the unique maximum w.r.t. $\preceq_T$. The set of inner vertices $V(T) \setminus L(T)$ is denoted by $V^0(T)$, and a subtree of T which is rooted at a vertex $u \in V^0(T)$ and contains all nodes and edges below u by $T(u)$. Likewise, $L(T(u))$ is the set of leaves under u.

A *leaf coloring* is a surjective map $\sigma\colon L(T) \to \mathcal{C}$ that assigns a color to the set of leaves of a tree. Given such a map, $L[r]$ refers to the set of leaves with color $r \in \mathcal{C}$; more formally $L[r] \coloneqq \{x \in L(T) \mid \sigma(x) = r\}$.

The last common ancestor $\mathrm{lca}(A)$ of a subset $A \subseteq V(T)$ is the smallest vertex $v \in V(T)$ (w.r.t. $\preceq_T$) such that $x \preceq_T v$ for all $x \in A$. For easier notation, let $\mathrm{lca}(x, y) \coloneqq \mathrm{lca}(\{x, y\})$.

A tree T *displays* another tree T' if T' can be obtained from a subtree of T by contraction of edges (in the usual sense). If not stated otherwise, the displayed tree for a restricted set of leaves L' of T means the tree that is obtained by subsequently removing all $v \in V(T) \setminus L'$ with $|\mathsf{child}(v)| = 0$ (including their incident edges) and contracting vertices $v \in V(T)$ with $|\mathsf{child}(v)| = 1$ and their two incident edges into a single edge. *Rooted triples* are a special kind of displayed trees. A rooted triple $xy|z$ is a rooted tree on three leaves x, y, z such that the path from x to y and the path from z to the root do not intersect. A rooted triple is called consistent with a tree T if $x, y, z \in L(T)$ and $\mathrm{lca}(x, y) \prec_T \mathrm{lca}(x, z) = \mathrm{lca}(y, z)$.

The term rooted triple is closely related to the definition of outgroups:

Definition 3 (Outgroup)**.** *A leaf $z \in L(T)$ is called an* outgroup *w.r.t. a set $X \subset L(T)$ if* $\mathrm{lca}(X) \prec \mathrm{lca}(X \cup \{z\})$.

In particular, T displays the rooted triple $x'x''|z$ for all distinct $x', x'' \in X$ and $z \in L(T) \setminus L(T(\mathrm{lca}(X)))$.

Finally, a tree T (and a graph in general) can be endowed with a map that assigns weights to its edges. In the case of positive weights, these can be interpreted as the lengths of the edges and represented by a weight function $\ell\colon E(T) \to \mathbb{R}^+$.

3.1.2 Phylogenetic Trees

This section describes the properties and the associated maps of phylogenetic trees, i.e., graph models of evolutionary histories of either species or genes.

Definition 4 (Phylogenetic tree [cf. 40, 28])**.** *An unrooted tree $\overline{T}$ is a* phylogenetic tree *if every inner vertex $v \in V^0(\overline{T})$ has a degree of at least 3. A rooted tree T is a* phylogenetic tree *if every $v \in V^0(T)$ has at least 2 children. $\overline{T}$ and T are called* fully-resolved *if the respective equalities hold, i.e., $\deg(v) = 3$ for every $v \in V^0(\overline{T})$ or $|\mathsf{child}(v)| = 2$ for every $v \in V^0(T)$, respectively.*
A planted *phylogenetic tree T is a rooted phylogenetic tree with a special vertex 0_T, called the* planted root*, that has a single child ρ_T such that $T(\rho_T)$ is a phylogenetic tree.*

The purpose of this definition is to avoid vertices of degree 2 (possibly with exception of the root). Such vertices would lack a justification by an evolutionary event that could be reconstructed from observable data with suitable methods. In contrast, a vertex that is not fully-resolved can be interpreted as missing information about the exact local topology in most cases. The planted edge $0_T\rho_T$ is useful in the wake of modeling events that predate the first branching event. This becomes especially relevant for the reconciliation of gene

trees with the underlying history of the corresponding species, since, e.g., duplication events can occur before the first speciation.

Exact methods for phylogenetic reconstruction cannot per se determine the location of the root. Hence, it is often necessary to consider unrooted trees. An unrooted version $\overline{T}$ of a rooted tree T with distance function ℓ can be obtained by the following two operations [74]:

(i) Omit the planted root 0_T and its incident edge.

(ii) In case the root ρ_T has exactly two children u_1 and u_2, replace the path $u_1 - \rho_T - u_2$ by a single edge u_1u_2 with length $\ell(u_1u_2) \coloneqq \ell(\rho_T u_1) + \ell(\rho_T u_2)$.

The weight function ℓ is the same for all other edges. However, note that the ancestor order $\preceq_T$ must be dropped.

As already mentioned, phylogenetic trees can either be species trees, that represent the relationship and branching history of different taxa, or gene trees, which constitute the history of a gene family. In the latter case, duplication and HGT events cause additional branching, whereas losses terminate existing branches. Since all members of a gene family reside in some species, there exists an embedding of a gene tree T into the corresponding species tree S.

In case of a gene tree T on a set of extant genes $L(T)$, the knowledge about which genes reside in which species is represented by a leaf coloring $\sigma\colon L(T) \to L(S)$. Therefore, the color and species of a leaf $v \in L(T)$ will be used as synonymous terms. The rest of the embedding can be formalized by a reconciliation map:

Definition 5 (Reconciliation Map [cf. 28, 74])**.** *Let $S = (W, F)$ and $T = (V, E)$ be two planted phylogenetic trees and let $\sigma\colon L(T) \to L(S)$ be a surjective map. A reconciliation from (T, σ) to S is a map $\mu\colon V \to W \cup F$ satisfying*

(R0) Root Constraint. *$\mu(x) = 0_S$ if and only if $x = 0_T$.*

(R1) Leaf Constraint. *If $x \in L(T)$, then $\mu(x) = \sigma(x)$.*

(R2) Ancestor Preservation. *If $x \prec_T y$, then $\mu(x) \preceq_S \mu(y)$.*

(R3) Speciation Constraints. *Suppose $\mu(x) \in W^0$.*

(i) $\mu(x) = \mathrm{lca}_S(\mu(v'), \mu(v''))$ for at least two distinct children v', v'' of x in T.

(ii) $\mu(v')$ and $\mu(v'')$ are incomparable in S for any two distinct children v' and v'' of x in T.

(R4) Speciation Constraint II. *If $\mu(\mathrm{lca}_T(x, y)) = \mu(\mathrm{lca}_T(x, z)) \in V^0(S)$, then $\mathrm{lca}_S(\sigma(x), \sigma(y)) = \mathrm{lca}_S(\sigma(x), \sigma(z))$ for all distinct leaves $x, y, z \in L(T)$.*

Axioms (R0) to (R4) hold for gene family histories that do not include horizontal gene transfer events. The first two axioms ensure that the root and leaves of T also map to the root and leaves in the species tree S, respectively. Note that the *Leaf Constraint* thereby implicitly prohibits loss leaves since they could not be mapped to an extant species. Thus, both σ and μ would be undefined for such kind of leaves. The *Ancestor Preservation* constraint forbids that a descendant x of some vertex y in T is mapped above $\mu(y)$ in S which avoids time traveling of a gene into an ancestor species. The two constraints in (R3) refer to restrictions for (observable) speciation events in the gene tree. In particular, (R3 ii) states that any two children v' and v'' of such a vertex are not only incomparable in T but also in S, i.e., $\mu(v')$ and $\mu(v'')$ are incomparable w.r.t. $\preceq_S$. This corresponds to a separated evolution of the gene family in the descending branches of a speciation event. Finally, axiom (R4) has been introduced only recently by Stadler et al. [74]. It forbids

to map two vertices $v_1, v_2 \in V^0(T)$ that represent distinct speciation events to the same vertex in S. This avoids ambiguous interpretations of single vertices as multiple events.

In case of the occurrence of HGT events, the axiom system has to be modified [28]. First, a weaker version of axiom (R2) is satisfied:

(R2*) Weak Ancestor Preservation. *If $x \prec_T y$, then either $\mu(x) \preceq_S \mu(y)$ or $\mu(x)$ and $\mu(y)$ are incomparable w.r.t. $\preceq_S$.*

Moreover, two additional axioms are necessary:

(R3) *(iii)* (Addition to the Speciation Constraints.) *Suppose $\mu(x) \in W^0$. If $\mu(x) \in W^0$, then $\mu(v) \preceq_S \mu(x)$ for all $v \in \mathsf{child}(x)$.*

(R5) HGT Constraint. *If x has a child y such that $\mu(x)$ and $\mu(y)$ are incomparable, then x also has a child y' with $\mu(y') \preceq_S \mu(x)$.*

This extended axiom system is a valid generalization in most cases. However, it fails in some scenarios, e.g., (R3 ii) may not hold if an HGT event has no surviving members in the non-transferred branch [cf. 28, Section 7.3].

An event labeling that is based on the reconciliation map is defined as follows:

Definition 6 (Event Labeling [28, Def. 3 with HGT])**.** *Given a reconciliation map μ from (T, σ) to S, the* event labeling on T (determined by μ) *is the map $t\colon V(T) \to \{\circledcirc, \odot, \bullet, \square, \triangle\}$ given by:*

$$t(u) = \begin{cases} \circledcirc & \text{if } u = 0_T, \text{ i.e., } \mu(u) = 0_S \text{ (root)} \\ \odot & \text{if } u \in L(T), \text{ i.e., } \mu(u) \in L(S) \text{ (leaf)} \\ \bullet & \text{if } \mu(u) \in V^0(S) \text{ (speciation)} \\ \square & \text{if } \mu(u) \in E(S)\text{, } u \text{ and } v \text{ are comparable for all } v \in \mathsf{child}(u) \text{ (dupl.)} \\ \triangle & \text{if } u \text{ has a child } v \text{ such that } \mu(u) \text{ and } \mu(v) \text{ are incomparable (HGT)} \end{cases}$$

Note that Definition 6 does not capture loss events. As they have shown to be an important cause of problems for inference methods, trees that additionally contain all branches leading to loss events will often be considered. Such complete gene family histories will be referred to as *extended gene trees*. Loss events will be indicated by $\dashv$ (or an asterisk $*$ in the implementation, reps.), whilst the symbols of the event labeling are used for all other event types. In this context, phylogenetic trees in the usual sense correspond to the observable part of the complete gene trees. Observable gene trees are easily constructed from the latter:

(i) Remove all branches leading to loss events only.

(ii) Subsequently, contract all nodes of degree 2 (except the root) and the adjacent edges into a single edge.

In ambiguous cases, the observable gene tree will sometimes be denoted by T_{obs}. An example of an (extended) gene tree T embedding is shown in Figure 3.1. It contains the different event types including losses.

Note that the root ρ_T is not a speciation event but a duplication in the example. Since it will be useful later, the following special type of duplication events is introduced:

Definition 7. *A duplication event $v \in V^0(T)$ is called* ancient *if v is mapped to the edge $0_S\rho_S$ under the reconciliation map μ.*

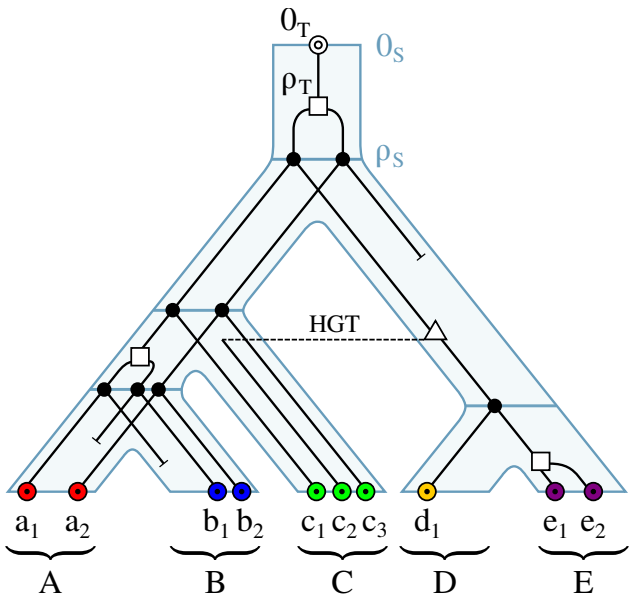

Figure 3.1: Embedding of an extended gene tree into the corresponding species tree. The species tree S on the set of extant species $\{A, B, C, D, E\}$ is shown in light blue. The gene family history includes speciation (●), duplication (□), HGT (△) and loss (⊣) events. The colors of the non-loss leaves (⊙) constitute known information about the extant genes, i.e., the species in which they reside.

It will sometimes be necessary to consider duplication events that predate the last common ancestor of a subset $L' \subseteq L(T)$. Ancient duplications w.r.t. $u = \mathrm{lca}_T(L')$ are defined by applying Definition 7 to the subtree $T(u)$ where the unique path between 0_T and u is contracted into a planted edge for $T(u)$.

Since all vertices in phylogenetic trees represent evolutionary events, it is useful to have a dating function τ that assigns a time point to every vertex. Following the conventions e.g. used by Böcker and Dress [5], these time points will be normalized such that $\tau(0_S) = 1$ and $\tau(x) = 0$ for all $x \in L(S)$. In case the tree is not planted, $\tau(\rho_T)$ is set to 1. Hence, the dating function maps to the unit interval: $\tau\colon V(T) \to [0, 1]$.

3.1.3 Metrics and Ultrametrics

Both the dating function $\tau\colon V(T) \to [0, 1]$ and the weight function $\ell\colon E(T) \to \mathbb{R}^+$ can be used to define a distance function on the set of vertices of a tree. Since the dating function assigns time points to the vertices of T, the time difference between two vertices x and y (that are comparable in T) can be interpreted as the time of divergence that lies between them. In general, the *divergence time* d_τ between two arbitrary nodes $x, y \in V(T)$ is given by

$$d_\tau(x, y) = \sum_{uv \in P} |\tau(u) - \tau(v)| \tag{3.1}$$

where P is the unique path between x and y. In contrast, the weighting function ℓ can be used to model different aspects of the evolutionary history. In particular, the weights can represent the dissimilarity or evolutionary distance of adjacent vertices in a gene tree T. Thus, the distance that is given by the sum of all edge weights on the unique path P

between two vertices $x, y \in V(T)$

$$d(x, y) = \sum_{e \in P} \ell(e) \tag{3.2}$$

corresponds to the *evolutionary distance* between x and y.

The set of vertices $V(T)$ together with one of the two distance functions d_τ or d forms a *metric space*. A metric is defined as follows:

Definition 8 (Metric & Ultrametric)**.** *A* metric *(also called* distance function*) on a set X is a map $d\colon X \times X \to \mathbb{R}_0^+$ such that for all $x, y, z \in X$, the following conditions are satisfied:*

(i) Non-negativity: $d(x, y) \geq 0$.

(ii) Identity of indiscernibles: $d(x, y) = 0 \iff x = y$.

(iii) Symmetry: $d(x, y) = d(y, x)$.

(iv) Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

An ultrametric *is a metric that satisfies a stronger version of the triangle inequality:*

(iv)* $d(x, z) \leq \max(d(x, y), d(y, z))$.

Since they represent the extant members of a species or gene family, the distances between the leaves of a tree are of special relevance. In particular, evolutionary distances between extant genes often constitute the available information as discussed in later sections.

For a normalized dating function $\tau : V(T) \to [0, 1]$ and two extant genes $x, y \in L(T)$ the divergence time simplifies to the map

$$d_\tau(x, y)\colon L(T) \times L(T) \to \mathbb{R}_0^+ \colon \quad (x, y) \mapsto 2\tau(\mathrm{lca}_T(x, y)) \tag{3.3}$$

which has the well-known property of representing a 1-to-1 correspondence between dated, rooted trees and ultrametrics [cf. 34, 5]. An *ultrametric tree*, i.e., a rooted tree with an arbitrary distance function having the property of ultrametricity, is most naturally shown visually by positioning all its leaves on the same level such that they have the same distance to the root (see Figure 3.2, left).

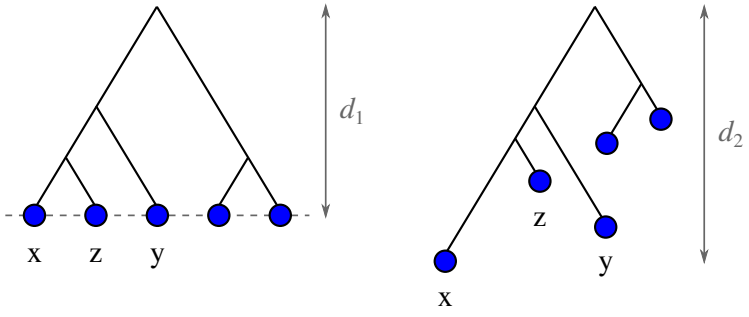

Figure 3.2: Ultrametric and non-ultrametric tree. The vertical components of the edges represent the distances. In the left tree, they induce an ultrametric on the set of its leaves, whereas they do not in the tree on the right side. This can, e.g., be seen by the violation of condition *(iv*)* in Definition 8 for the leaves x, y, z: $d_2(x, y) > \max(d_2(x, z), d_2(z, y))$.

In contrast, the distance function d on the set of leaves $L(T)$ is not an ultrametric on the set of leaves in general. This is especially the case for evolutionary distances in the presence of asymmetric divergence as it will be discussed later. However, the distance function d satisfies a weaker property by construction: A metric d on set X is *additive* if a (not necessarily rooted) weighted tree (T, ℓ) exists such that $L(T) = X$ and $d = d_T$ where d_T is the distance function of (T, ℓ) as defined by Equation 3.2. A metric on a set X can be tested for additivity with the four-point-condition: For any *quartet*, i.e., a set of four nodes $x, y, u, v \in X$, it must hold that out of the three distance sums

(i) $d(x, y) + d(u, v)$,

(ii) $d(x, u) + d(y, v)$,

(iii) $d(x, v) + d(y, u)$

two sums are equal and not smaller than the third [cf. 71, 11].

Definition 9 (Quartet Relation [cf. 74])**.** *Consider an unrooted tree $\overline{T}$ with leaf set $L(\overline{T})$. For any four distinct leaves $x, y, u, v \in L(\overline{T})$ denote by $\overline{T}[x, y, u, v]$ the unrooted tree obtained by suppressing all vertices of degree 2 in the union of the paths in $\overline{T}$ that connect x, y, u, v. The* quartet relation *for $\overline{T}$ is*

(i) $\overline{T}[x, y, u, v] = (xy|uv)$,

(ii) $\overline{T}[x, y, u, v] = (xu|yv)$ *or*

(iii) $\overline{T}[x, y, u, v] = (xv|yu)$

if there is an edge $e \in E(\overline{T})$ such that the respective pairs (that are separated by the vertical bar) are in different connected components of $\overline{T}$ after the removal of e. If there is no such edge write $\overline{T}[x, y, u, v] = \times$.

The four-point-condition is directly related to this notion of quartets: For an additive metric d, the smallest of the three distance sums induced by four distinct points x, y, u and v determines their topology in a corresponding tree $\overline{T}$ that explains d. The reason for this is that the edge separating two pairs in $\overline{T}[x, y, u, v]$ only contributes to the two larger sums. The case in which all three sums are equal corresponds to the absence of such a separating edge in $\overline{T}$ (and hence also in $\overline{T}[x, y, u, v]$) and is referred to as *star tree* or *star topology*. The possible cases are visualized in Figure 3.3.

Given a rooted or unrooted tree T and a unique numbering of the leaves $L(T)$, the distance function on the leaves can be represented by a symmetric square matrix which will be denoted by $\mathbf{D}$ in the following, where $\mathbf{D}(x, y)$ is the entry in the row and column corresponding to x and y, respectively.

3.1.4 Homology and Best Matches

As already outlined, the type of relationship between pairs of genes is of interest for the inference of gene functions. In particular, orthologous genes are considered to perform similar functions. Mathematically, both orthology and paralogy are binary relations on the set of (extant) genes in a gene tree T. They are defined w.r.t. the event type of the last common ancestor of two genes x and y:

Definition 10 (Orthology and Paralogy [24])**.** *Let (T, μ, t) be an event-labeled, rooted gene tree with reconciliation map μ. Two distinct leaves $x, y \in L(T)$ are* orthologs w.r.t. *μ if $t(\mathrm{lca}_T(x, y)) = \bullet$, and* paralogs *if $t(\mathrm{lca}_T(x, y)) = \square$.*

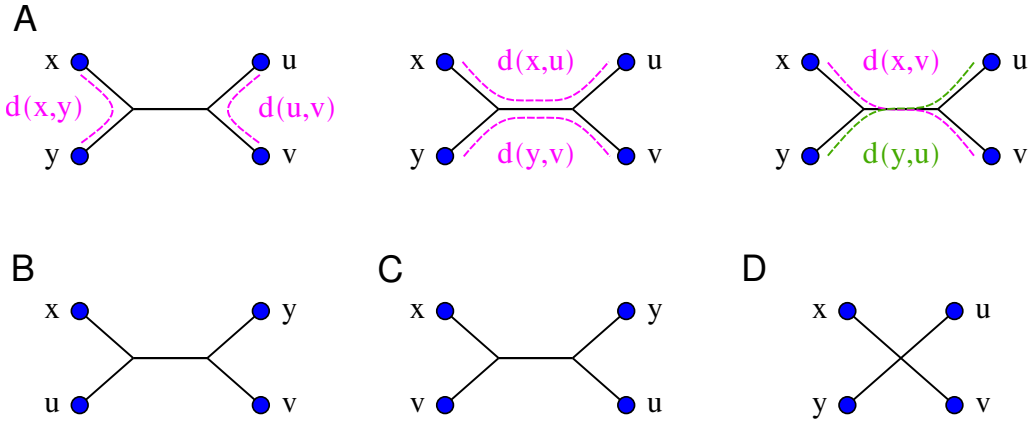

Figure 3.3: The four possible (unrooted) quartets. For case (A), the six distances that form the three distance sums are indicated. In formal, the cases are:

(A) $\overline{T}[x,y,u,v] = (xy|uv)$
$\iff d(x,y) + d(u,v) < d(x,u) + d(y,v) = d(x,v) + d(y,u)$

(B) $\overline{T}[x,y,u,v] = (xu|yv)$
$\iff d(x,u) + d(y,v) < d(x,y) + d(u,v) = d(x,v) + d(y,u)$

(C) $\overline{T}[x,y,u,v] = (xv|yu)$
$\iff d(x,v) + d(y,u) < d(x,y) + d(u,v) = d(x,u) + d(y,v)$

(D) $\overline{T}[x,y,u,v] = \times$
$\iff d(x,y) + d(u,v) = d(x,u) + d(y,v) = d(x,v) + d(y,u)$

Of course, the true reconciliation map and event-labeling are not known for real-life data. Therefore, methods exist that aim to explicitly reconstruct the gene tree and the reconciliation map, but also others that attempt to directly infer orthology based on gene similarity. In both cases, a well-founded mathematical investigation of the orthology and paralogy relation is helpful.

Firstly, Defintion 10 unambiguously defines two distinct members of a gene family as either orthologs or paralogs in the absence of horizontal gene transfer. Otherwise, there may be pairs of genes $x, y \in L(T)$ having a horizontal gene transfer as last common ancestor, and, thus, $t(\mathrm{lca}_T(x,y)) = \triangle$. Clearly, x and y are neither orthologs nor paralogs given the definition above. Therefore, *xenology* was introduced as a third variant of homology. There are several definitions of xenologs. Fitch [24] calls two genes xenologs if there is at least one HGT event on the unique path in T connecting them. Thus, it is not necessary that their last common ancestor was an HGT event. As a consequence, two genes can be both xenologs and ortho-/paralogs. They are commonly termed as *xeno-orthologs* and *xeno-paralogs* in this case. An alternative definition by Hellmuth and Wieseke [36] avoids this ambiguity by calling two $x, y \in L(T)$ *(lca-)xenologs* if and only if $t(\mathrm{lca}_T(x,y)) = \triangle$. Analogously, *(lca-)orthologs* and *(lca-)paralogs* are defined. If not declared otherwise, all three terms will refer to this definition in the following.

Orthology and paralogy are irreflexive and symmetric binary relations, since for any $x \in L(T)$, it holds that $t(\mathrm{lca}_T(x,x)) = t(x) = \odot$, and for any two genes $x, y \in L(T)$, $t(\mathrm{lca}_T(x,y)) = t(\mathrm{lca}_T(y,x))$. Hence, they can be represented by undirected graphs. However, they are not transitive, as the example in Figure 3.4 shows (consider, e.g., the three genes a_1, b_1, c_1 in the orthology graph Θ).

Definition 11 (Orthology Graph [28, Def. 5 mod.])**.** *Let T be a gene tree with event labeling t. Let Θ be the undirected graph on $L(T)$ with $xy \in E(\Theta)$ if and only if* $\mathrm{lca}_T(x,y) =$

●. *Then* Θ *is called an* orthology graph *that is explained by the orthology relation of* (T, t).

Similarly, the paralogy graph $\overline{\Theta}$ is defined. In the absence of HGT events, it is simply the complement of Θ.

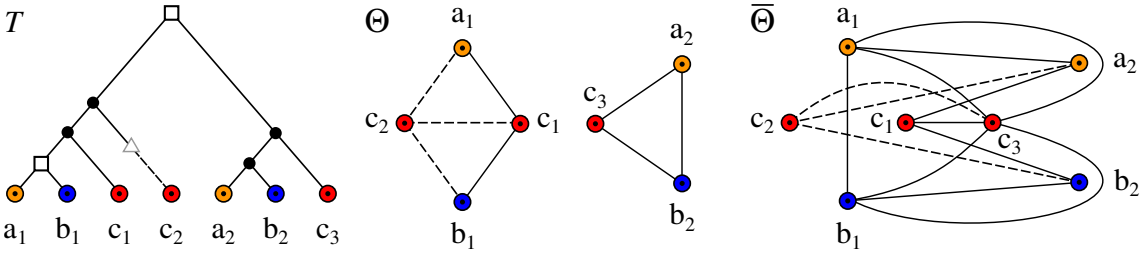

Figure 3.4: Corresponding orthology graph Θ and paralogy graph $\overline{\Theta}$ for the example gene tree in Figure 3.1 restricted to the set of species $\{A, B, C\}$ (see tree T on the left). Xeno-orthologous and -paralogous relations, respectively, are indicated by dashed lines.

Neither the orthology nor the paralogy relation can be observed directly. Therefore, heuristics have been developed that make use of the fact that orthologs are often the most closely related genes in two species. This approach has been used widely in phylogenetic reconstruction methods and requires some definition of the relationship between genes. To this end, different terms and variants have been introduced: Symmetric best matches [e.g. used by 75], bidirectional best hits (BBH) [e.g. 62, 53], reciprocal best hits (RBH) [e.g. 6], reciprocal best alignment heuristic (RBAH) [e.g. 54] and some others.

We strictly distinguish between *best hits* and *best matches*:

Definition 12 ((Reciprocal) Best hit)**.** *Consider a gene tree* T *with leaf set* $L(T)$*, a surjective map* $\sigma\colon L(T) \to L(S)$ *(where* $L(S)$ *is the corresponding set of species) and a distance function* $d : L(T) \times L(T) \to \mathbb{R}_0^+$*. Then* $y \in L(T)$ *is a* best hit *of* $x \in L(T)$ *if and only if* $d(x, y) \leq d(x, y')$ *holds for all leaves* y' *from species* $\sigma(y') = \sigma(y)$*.*
If x *is also a best hit of* y*,* x *and* y *are called* reciprocal best hits*.*

Thus, *best hits* are defined in the context of the smallest evolutionary distance. On the other hand, *best matches* refer to the closest relatives of a gene w.r.t. the point in time when they were separated:

Definition 13 ((Reciprocal) Best match [27])**.** *Consider a gene tree* T *with leaf set* $L(T)$ *and a surjective map* $\sigma\colon L(T) \to L(S)$ *(where* $L(S)$ *is the corresponding set of species). Then* $y \in L(T)$ *is a* best match *of* $x \in L(T)$*, in symbols* $x \to y$*, if and only if* $\mathrm{lca}(x, y) \preceq \mathrm{lca}(x, y')$ *holds for all leaves* y' *from species* $\sigma(y') = \sigma(y)$*.*
If x *is also a best match of* y *in color* $\sigma(x)$*, i.e.,* $y \to x$*,* x *and* y *are called* reciprocal best matches*.*

A comprehensive mathematical theory on best matches was developed only recently by Geiß et al. [27]. Therein, the authors point out that best hits and best matches are the same if the mutation rate of the gene family is constant, i.e., the *Molecular Clock* holds.

The best match relation can be represented by a directed, vertex-colored graph on the set of leaves of a gene tree T as follows:

Definition 14 (cBMG and cRBMG [27])**.** *Given a gene tree* T *and a map* $\sigma : L(T) \to L(S)$*, the* colored Best Match Graph (cBMG) $\vec{G}(T, \sigma)$ *has vertex set* $L(T)$ *and arcs* $(x, y) \in$

$E(\vec{G})$ if $x \neq y$ and $x \to y$. Each vertex $x \in L(T)$ obtains the color $\sigma(x)$.
The rooted tree T explains *the vertex-colored graph $(\vec{G}, \sigma)$ if $(\vec{G}, \sigma)$ is isomorphic (in the usual sense, with preservation of colors) to the cBMG $\vec{G}(T, \sigma)$.*
The vertex-colored undirected graph $G(T, \sigma)$ that has vertex set $L(T)$ and edges $xy \in E(G)$ if $x \neq y$ and $x \to y$ as well as $y \to x$ is called the colored Reciprocal Best Match Graph (cRBMG).

Geiß et al. [27] present two polynomial-time algorithms to decide whether a given digraph $(\vec{G}, \sigma)$ is a valid colored Best Match Graph and to determine the unique least resolved tree, i.e., the corresponding tree which explains $(\vec{G}, \sigma)$ and is minimal in the sense that no edge can be contracted such that the tree still explains $(\vec{G}, \sigma)$. The algorithm for computing the least resolved tree that will be relevant later in this contribution makes use of informative triples:

Definition 15 (Informative triples [27, Def. 8])**.** *Let $(\vec{G}, \sigma)$ be a two-colored digraph. We say that a triple $ab|c$ is* informative *(for $(\vec{G}, \sigma)$) if the three distinct vertices $a, b, c \in L$ induce a colored subgraph $\vec{G}[a, b, c]$ isomorphic (in the usual sense, with preservation of colors) to the graph X_1, X_2, X_3 or X_4 shown in Figure 3.5. The set of informative triples is denoted by $\mathcal{R}(\vec{G}, \sigma)$.*

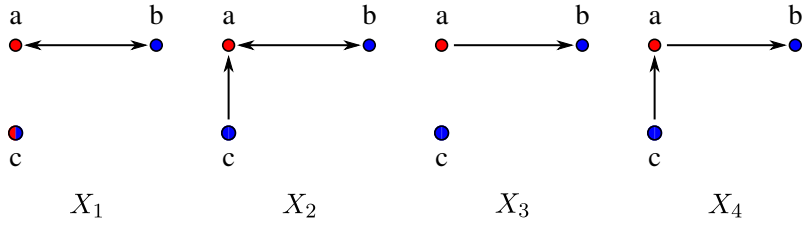

Figure 3.5: Each of the three-vertex induced subgraphs X_1, X_2, X_3 and X_4 gives a triple $ab|c$. If vertex c in the drawing has two colors, then the color $\sigma(c)$ does not matter [27].

This definition can easily be extended for n-colored graphs. Geiß et al. [27] shows that the well-known polynomial-time algorithm `BUILD` [1] can be used to compute the least resolved tree directly from the full set of informative triples $\mathcal{R}(\vec{G}, \sigma)$ of an n-colored Best Match Graph. Moreover, a colored digraph $(\vec{G}, \sigma)$ is a valid cBMG if and only if $(\vec{G}, \sigma) = \vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$ where $\mathrm{Aho}(\mathcal{R})$ denotes the tree T that results from applying `BUILD` to a set of triples $\mathcal{R}$ [27, cf. Theorems 6 and 9]. In other words, the Best Match Graph of the resulting tree has to be equal to the original digraph. Therefore, the explicit construction of a least resolved tree can be used to decide whether a given digraph is a cBMG.

In a subsequent publication, the properties of *colored Reciprocal Best Match Graphs (cRBMG)* have been studied in detail [29]. Two genes x and y are reciprocal best matches if and only if both x is a best match of y and y is a best match of x. As mentioned above, the cRBMG (possibly with some corrections based on the corresponding cBMG) is a good heuristic for the true orthology relation. In fact, given a tree T with reconciliation map μ and a corresponding event-labeling t_μ, the orthology graph $\Theta(T, t_\mu)$ is a subgraph of the Reciprocal Best Match Graph $G(T, \sigma)$ [28, Theorem 2] in the absence of horizontal gene transfer. Thus, the edges of G cannot contain false positives w.r.t. the orthology relation.

Figure 3.6 shows the cBMG $\vec{G}$, the cRBMG G and the true orthology relation Θ that correspond to the previous example tree. The orthology graph is a subgraph of the cRBMG with exception of the xeno-orthologous relations. The simulations by Geiß et al. [28] show that reciprocal best matches are a useful heuristic for orthology also if HGT occurs (at moderate rates).

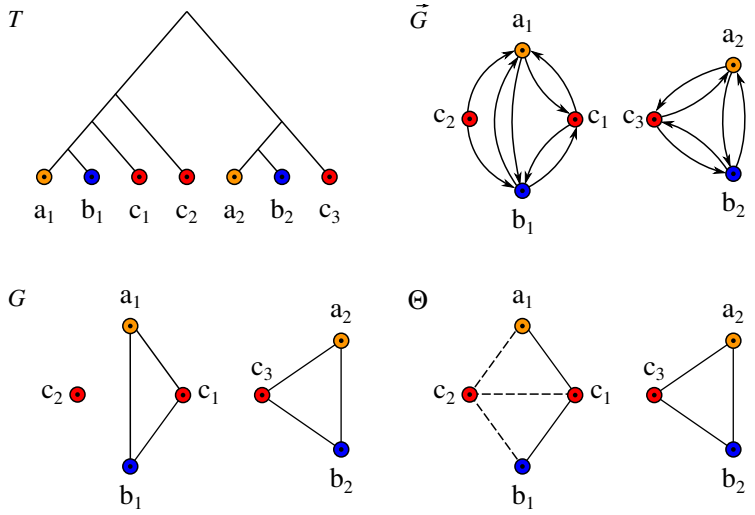

Figure 3.6: Gene tree with corresponding colored Best Match Graph $(\vec{G}, \sigma)$ and Reciprocal Best Match Graph (G, σ). The tree is a subtree of the gene tree in Figure 3.1 restricted to the set of species $\{A, B, C\}$. The true orthology graph Θ is again depicted for comparison.

However, especially in case of multiple losses, the true orthology relation is often hard to infer. The example includes the well-known problematic case of differential gene loss after a dupiclation event followed by speciation. The last common ancestor of genes a_1 and b_1 is a duplication (see Figure 3.1). The two loss events cause a false-positive edge when considering G as the estimated orthology relation.

3.2 Simulation of Gene Family Histories

In the first two sections of this chapter, the methods for the simulation of dated species and gene trees are described. They are based on the methods used by Hernandez-Rosales et al. [41] and López Sánchez [56]. They generate the gene trees by constructing Poisson vectors of events for the edges of the species tree S, whereas a variant of the Gillespie algorithm [31] is used here.

In the rest of the chapter, the simulation of evolution rate imbalances and measurement noise is discussed. The former is a result of the divergent fates of genes after their separation by speciation, duplication or horizontal gene transfer, and is therefore commonly observed in biological data. For the distances of the leaves of T, this destroys the property of ultrametricity. An overview of the simulation framework is supplied in Figure 3.7.

Moreover, in Fig. 3.8, an example for a simulated scenario is shown including a species tree, an imbalanced gene tree and the corresponding observable gene tree, i.e., the gene tree after removing all branches that lead to loss events only and then suppressing all inner vertices that have only one child.

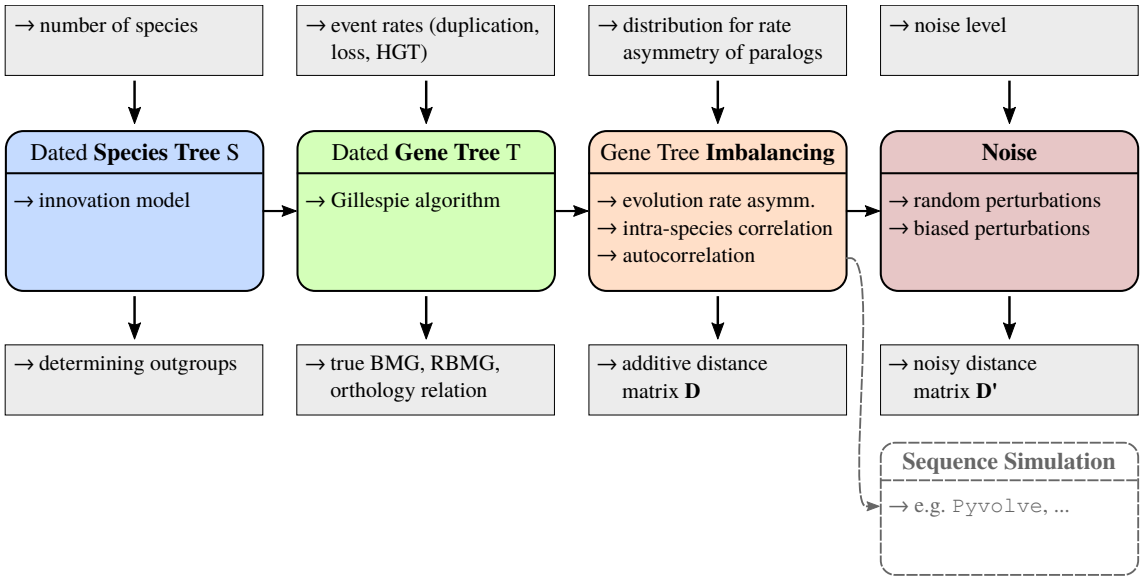

Figure 3.7: Overview of the framework for the simulation of realistic evolutionary distance data. The four main modules are indicated by the colored boxes, the most important input as well as some additional output data above and below, respectively.

3.2.1 Simulation of Species Trees

The *innovation model* described by Keller-Schmidt and Klemm [47] produces tree topologies that are comparable with those of real phylogenetic trees considering properties like average leaf depth and tree imbalance. It assumes that that evolutionary branching is driven by innovations, i.e., the generation of novel features or the loss of such features.

In order to produce realistic gene family histories, the species tree S has to meet some additional requirements. Firstly, it is possible that events, such as gene duplications or losses, predate the first speciation event in biological gene trees. Therefore, an additional node 0_S is added to S and connected to the first speciation event ρ_S. Hence, in the following generation of gene trees T along the planted species tree S, events can already occur within the edge $0_S\rho_S$.

Secondly, a dating function $\tau : V(S) \to [0, 1]$ has to be constructed to introduce a realistic variance for the time intervals between speciation events and to determine the set of available edges for a given time τ (especially in the wake of HGT). For the construction of the dating function τ, the tree is traversed in pre-order and the following rules are applied: For each $v \in V(S)$, set $\tau(v) = 1$ if $v = 0_S$, $\tau(v) = 0$ if v is a leaf. Otherwise, pick a random leaf $l \in L(S(v))$ and determine the length of the path P (i.e., the number of edges $|P|$) from v to l. Next, a random number r is drawn from a uniform distribution on the interval $[0, 2]$. It is used to introduce more variability into the lengths of the edges by setting $\tau(v) = \tau(u) \cdot r/(|P| + 1)$. In this equation, u is the parent of v. The pre-order traversal ensures that $\tau(u)$ is already set for any $v \in V(S) \setminus \{0_T\}$. The procedure is summarized in Algorithm 1.

Note that the algorithm also works for unplanted trees which are normalized such that $\tau(\rho_S) = 1$. The length of each edge $uv \in E(S)$ in the resulting dated species tree (in the sense of divergence time) is given by $\tau(u) - \tau(v)$. Furthermore, (S, τ) induces an ultrametric on the set of leaves $L(S)$ as described in Section 3.1.3.

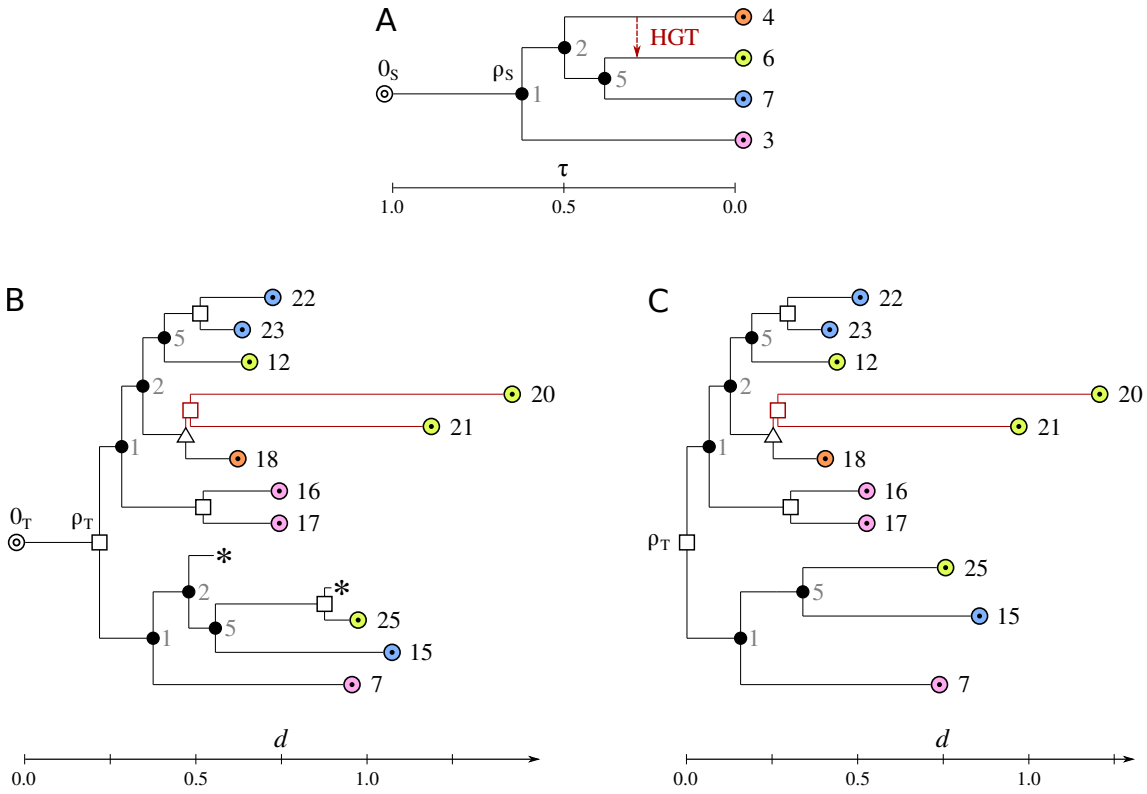

Figure 3.8: Example simulated species tree and imbalanced gene tree. (A) Planted species tree on four (extant) species. The dating function is represented by the scale below. (B) Extended gene tree, i.e., the complete history of the simulated gene family including speciations (●), duplications (□), losses (∗) and an HGT event (△; transferred branch in red). The numbers at the speciation events correspond to the numbering in (A), the numbers at the leaves are their unique IDs. The leaf coloring indicates the corresponding species. The scale represents the evolutionary distance from the root 0_T. The lengths of the horizontal lines are all to scale. (C) Observable gene tree.

3.2.2 Simulation of Gene Trees

The Gillespie Algorithm

The history of a gene family is embedded into the phylogeny of the corresponding species tree. Mathematically, this relationship is formalized in the reconciliation map $\mu : V(T) \to V(S) \cup E(S)$ which was described in Section 3.1.2. Whenever a speciation event, resulting in two separated branches in S, occurs, the evolutionary histories of all genes in the ancestral species are also separated. Additionally, duplication, horizontal gene transfer, and loss events occur with certain rates during the evolution of the species, which also lead to branching in the gene tree T or termination of branches in the latter case.

Whilst the speciation events are fixed by S, the Gillespie algorithm is a suitable choice for the exact stochastic simulation of events occurring with rates that may depend on the current state of the system. First introduced by Daniel Gillespie in 1976, the algorithm is best known for the simulation of chemical reaction kinetics in systems with relatively small numbers of molecules. Traditional methods that assume the number of reactants to be continuous fail in this situation [32].

In the original description [31], the overall simulation algorithm consists of the following

Algorithm 1: Construction of the dating function for S

Data: (Optionally planted) species tree S
Result: Dating function τ.

```
foreach v ∈ V(S) in pre-order do
    if par(v) = ∅ (root) then
        Set τ(v) = 1.
    else if |child(v)| = 0 (leaf) then
        Set τ(v) = 0.
    else
        Draw a random leaf l ∈ L(S(v)).
        Determine the length of the path P between v and l.
        Draw r ∈ [0, 2] uniformly.
        Set τ(v) = τ(par(v)) · r/(|P| + 1).
    end
end
```

(here slightly simplified) steps:

- *Step 1: Initialization*
  Initialize the time to $\tau = 0$ and the molecule numbers $X_1, X_2, ..., X_N$ for a set of N chemical species. Furthermore, specify the rates $r_1, r_2, ..., r_M$ for a set of M possible reactions $(\xi_1, \xi_2, ..., \xi_M)$ between these reactants. The rates are usually functions of the $X_1, X_2, ..., X_N$. Determine a stopping time τ_{stop}.

- *Step 2: Monte-Carlo-Step*
  Generate a random pair $(\Delta\tau, \mu)$ according to the joint probability function $P(\Delta\tau, \mu)$ where $\Delta\tau$ is the waiting time until the next event and $1 \leq \mu \leq M$ an integer deciding which of the M reactions will occur next.

- *Step 3: Update*
  Advance τ by $\Delta\tau$. Update the values of the chemical species $X_1, X_2, ..., X_N$ according to one occurrence of reaction ξ_μ as well as the reaction rates $r_1, r_2, ..., r_M$.

- *Step 4: End of iteration*
  Terminate the simulation if $\tau > \tau_{\text{stop}}$ or no more reactions can occur, i.e., all rates $r_1, r_2, ..., r_M$ are zero. Otherwise, return to Step 2.

For the correct generation of $\Delta\tau$ and μ in the Monte-Carlo-Step, Gillespie proposes two methods which he proofs to be equivalent: the 'Direct' and the 'First reaction' method. In the 'Direct' method, a total rate R is computed for each iteration step as

$$R = \sum_{\mu=1}^{M} r_\mu. \tag{3.4}$$

Next, two random numbers x_1 and x_2 are drawn from a uniform distribution over the interval $(0, 1]$ and $\Delta\tau$ is set to

$$\Delta\tau = (1/R) \ln(1/x_1). \tag{3.5}$$

Finally, the reaction ξ_μ is selected such that

$$\sum_{\nu=1}^{\mu-1} r_\nu < x_2 R \leq \sum_{\nu=1}^{\mu} r_\nu. \tag{3.6}$$

For the 'First reaction' method, individual waiting times $\Delta\tau_\nu$ are drawn for each of the M reactions again using random numbers x_ν and setting

$$\Delta\tau_\nu = (1/r_\nu)\ln(1/x_\nu) \qquad (\nu = 1, 2, ..., M). \tag{3.7}$$

Thus, a random number x_ν has to be drawn for every reaction. The next reaction ξ_μ with the corresponding waiting time $\Delta\tau_\mu$, that is actually carried out, is then chosen such that $\Delta\tau_\mu$ is smallest among the $\Delta\tau_1, \Delta\tau_2, ..., \Delta\tau_M$.

Gillespie [31] shows that both approaches are equivalent and correct implementations of the Monte-Carlo-Step w.r.t. the probability function $P(\Delta\tau, \mu)$. In both cases, the waiting time $\Delta\tau$ follows an exponential distribution with rate R as given by Equation 3.4. However, for the simulation of events along a species tree, some additional considerations have to be made which are described in the next section.

The Gillespie algorithm has previously been used for the simulation of gene family histories, e.g., in the Artificial Life Framework [ALF, 19], in which the set of possible events comprises duplications and losses, but also indels and substitutions in DNA sequences. This way, the evolution on the genome and sequence level is simulated in parallel.

Implementation

The Gillespie algorithm was originally designed for non-branching processes. For example, in the simulation of chemical reactions, the assumption is made that all molecules are contained in a defined volume and are therefore not restricted from interacting with one another [32]. In gene family histories, on the other hand, genes are separated by branching events such as speciations and duplications. The simulation procedure must therefore also enable branching in some way. It appears, in particular, that the order in which single branches are handled is relevant whenever dependencies exist, i.e., as well as chemical reactions in the original version, events can result in changes of rates.

Especially, the extinction of a whole gene family in a species should be avoided. It is thus necessary to keep track of the current number of genes in all species and set the loss rate $l_g = 0$ whenever gene g is the only copy in its species. Likewise, duplications and horizontal gene transfers (HGTs) can reset l_g, e.g., to some initially specified default value in the simplest case.

In this section, a variant of the Gillespie algorithm is described that is able to correctly handle the above-described cases. Moreover, it can be interpreted as a more stringent application of the 'Direct' method as compared to an alternative implementation in the next section. It is therefore also referred to as the *Direct method* here.

As a first modification to the original algorithm by Gillespie [31], the set of possible reactions is not fixed, but it is updated after every event instead. The reason for this is that all combinations of extant genes at a given time τ and event types are considered as the reactions, i.e., they are given by the Cartesian product $G(\tau) \times Q$ where $G(\tau)$ is the set of *active* genes, i.e., every $g \in G(\tau)$ is an extant gene at time τ, and $q \in Q := \{D, L, H\}$ is one of the three events duplication, loss and HGT. Hence, the reactions $\xi := (g, q)$ correspond to the possible simulation events excluding speciations. Furthermore, they are associated with a rate $r_\xi(\tau)$ that depends on the time τ. In particular, for the default settings, $r_\xi(\tau)$ will either be 0 or l if $q = L$, where l is a user-defined default loss rate. If $q \in \{D, H\}$ on the other hand, the rate will be equal to some fixed duplication rate d or HGT rate h, respectively. For each time point τ, the total event rate $R(\tau)$ is thus given by

$$R(\tau) = \sum_{\xi \in G(\tau) \times Q} r_\xi(\tau). \tag{3.8}$$

The simulation is initialized by setting $\tau = 1$ and $\tau_{\text{stop}} = 0$. Recall that $\tau(0_S) = 1$ and that the time decreases as the simulation progresses towards the leaves. All speciation events are kept in a priority queue $\mathcal{P}$ in temporal order. This is possible because they are known *a priori* from the dated species tree S. A single gene g_0 is associated with the planted edge $0_S\rho_S$ in S and added to the set of active genes. The total event rate is computed as $R(1) = d + h$, since losses are not allowed for a single gene. Note that, effectively, HGT events also cannot yet be carried out, since there is only a single species (see below).

In each iteration, first, a random number $\Delta\tau$ is drawn from an exponential distribution with rate parameter $R(\tau)$ representing the waiting time until the next duplication, loss or HGT event. Next, it is checked whether the interval $[\tau, \tau - \Delta\tau]$ contains the time point τ_s of the first element in the queue $\mathcal{P}$ containing the speciation events. If this is the case, the speciation is given priority and removed from the queue. The time τ is hence updated to τ_s and gene copies $g_1, ..., g_n$ of *each* parental gene g are placed into the respective edges $e_1, ..., e_n$ below the speciation event in S (where n is the number of children of the corresponding vertex in S). The copies $g_1, ..., g_n$ are associated with the event rates of the respective parental gene and added to the set of active genes, whereas the parental genes are removed from it.

Since the number of active genes has increased and thus also $R(\tau)$, a new waiting time $\Delta\tau'$ until the next event has to be drawn in the next iteration. This is the correct way of handling the situation, since the stochastic process of Gillespie's algorithm is essentially a first order Markov process [12], i.e., all probabilities solely depend on the current state of the system and are independent from the past. In particular, the distribution of waiting times from a time point τ_0 until the next event conditioned on the fact that no event occurred between τ_0 and τ_s, where τ_0 designates the time before it was updated to τ_s, can be expressed as follows: Consider the complementary cumulative distribution function and let $\sigma \coloneqq \tau_0 - \tau_s$:

$$\begin{aligned} \mathbb{P}(T \geq \sigma + \Delta\tau' | T \geq \sigma) &= \mathbb{P}(T \geq \sigma + \Delta\tau' \wedge T \geq \sigma)/\mathbb{P}(T \geq \sigma) \\ &= \mathbb{P}(T \geq \sigma + \Delta\tau')/\mathbb{P}(T \geq \sigma). \end{aligned} \tag{3.9}$$

The waiting time distributions are exponential with rate $R_1 \coloneqq R(\tau_0)$ before τ_s and rate $R_2 \coloneqq R(\tau_s)$ following the speciation event. Moreover, the event (in the sense of the sample space) that nothing happened in the time interval $[\tau_s, \tau_s - \Delta\tau']$ is independent from the event that nothing happened in $[\tau_0, \tau_0 - \sigma]$. Hence, one obtains

$$\mathbb{P}(T \geq \sigma + \Delta\tau' | T \geq \sigma) = \frac{\exp\left[-R_1\sigma\right] \exp\left[-R_2\Delta\tau'\right]}{\exp\left[-R_1\sigma\right]} = \exp\left[-R_2\Delta\tau'\right]. \tag{3.10}$$

The distribution of the waiting time after the speciation event is thus equal to an exponential distribution with rate parameter R_2. It is the same that will be used in the following iteration.

In case there is no speciation event interfering with the time interval $[\tau, \tau - \Delta\tau]$, the time is updated according to $\tau \leftarrow \tau - \Delta\tau$. If $\tau < \tau_{\text{stop}} = 0$, the iteration is ended. Otherwise, a second random number x is drawn from a uniform distribution on the unit interval and is then used to select the upcoming reaction, i.e., the active gene and event type. The probability for each reaction is equal to $r_\xi(\tau)/R(\tau)$ and the implementation follows the suggestion in the original algorithm (see Equation 3.6). The iteration proceeds depending on the selected gene g and event type q:

- *Duplication ($q = D$)*
  A gene duplication is modeled by placing a new copy g' into the same branch of S, i.e., the edge $e \in E(S)$ that embeds g, and adding g' to the set of active genes.

- *Loss ($q = L$)*
  A gene loss is modeled by removing g from the set of active genes.

- *HGT ($q = H$)*
  A horizontal gene transfer is modeled by choosing a target edge e' uniformly from all coexisting branches in S at time $\tau - \Delta\tau$ (excluding the branch in which g resides). If such an edge does not exist, nothing is done. Otherwise, a copy g' is placed into edge e' and added to the set of active genes.

In general, the rates $\xi_{(g,q)}(\tau - \Delta\tau)$ and $\xi_{(g',q)}(\tau - \Delta\tau)$ are set to $\xi_{(g,q)}(\tau)$. However, in the wake of restricting the extinction of the gene family, some special cases have to be considered:

- *Setting the loss rate to zero*
  If a loss event is performed leaving gene g^* as the last survivor in its species, then set $\xi_{(g^*,L)}(\tau - \Delta\tau) = 0$.

- *Resetting the loss rate*
  (i) If a duplication is performed, where g was the last survivor in its species, then reset $\xi_{(g,L)}(\tau - \Delta\tau) = \xi_{(g',L)}(\tau - \Delta\tau) = l$ where l is the user-defined default loss rate. (ii) If a HGT is performed, where g was the last survivor in its species, then reset $\xi_{(g,L)}(\tau - \Delta\tau) = l$. In any case, set $\xi_{(g',L)}(\tau - \Delta\tau) = l$ for HGTs.

The simulation is finished as soon as $\tau < \tau_{\text{stop}} = 0$. The construction of the dated gene tree T is done in parallel, i.e., vertices for every type of event are added, associated with the corresponding time stamp and connected to the correct parent node. After the last iteration, a leaf l_g with time stamp $\tau(l_g) = 0$ is added to T for all active genes $g \in G(0)$. Also, the reconciliation map between T and S is saved as a by-product. The complete simulation is summarized in Algorithm 2.

Algorithm 2: Direct Gillespie algorithm for the simulation of T

Input: Planted species tree S, default event rates d, l and h
Output: (Extended) Gene tree T and reconciliation map μ
Initialize gene tree T with root 0_T and reconciliation map μ.
Set $\tau = 1$ and $\tau_{\text{stop}} = 0$.
Initialize a priority queue P with the speciation events in temporal order.
Initialize set of active genes G with a single copy in the planted root of S, and the total rate $R = d + h$.
while $\tau > \tau_{stop}$ **do**
    Draw $\Delta\tau$ from an exponential distribution with rate $R(\tau)$.
    if $\tau - \Delta\tau < \tau_s$ *where* τ_s *is the time of the next speciation event* **then**
        Set $\tau = \tau_s$.
        Pop the next speciation event s at the front of P.
        Add copies $g_1, ..., g_n$ for *all* parental genes g into the corresponding edges below s and into G.
        Remove the parental genes g from G.
        Add a speciation vertex v for every parental gene g and the edge uv from its ancestor u to T.
    else if $\tau - \Delta\tau < \tau_{stop}$ **then**
        Set $\tau \leftarrow \tau - \Delta\tau$.
        continue
    else
        Set $\tau \leftarrow \tau - \Delta\tau$. Draw a random number r from a uniform distribution on $[0, 1]$.
        Select reaction $\xi = (g, q)$ using r.
        if $q = D$ **then**
            Add a copy g' of g into the same branch of S and to G.
        else if $q = L$ **then**
            Remove g from G.
        else if $q = H$ **then**
            if *another branch exists in* S **then**
                Select a coexisting branch in S and add a copy g' of g into it and into G.
            else
                continue
            end
        Add a vertex v for the event to T and the edge uv from its ancestor u.
    end
    Update all rates $r_\xi(\tau)$ and $R(\tau)$.
    Update the reconciliation map μ.
end
Add a leaf l_g for every gene $g \in G$ to T, and the edge ul_g from its ancestor u.

3.2.3 Divergence Asymmetries

Motivation

In Section 3.2.2, the simulation of dated gene trees has been described. As already introduced in the preliminaries section, the normalized dating function $\tau : V(T) \to [0, 1]$ induces an ultrametric divergence time d_τ between the non-loss leaves of the gene tree which is given by

$$d_\tau(x, y) = 2\tau(\mathrm{lca}_T(x, y)). \tag{3.11}$$

For biological data, divergence times cannot be observed directly in most cases. Similarities or dissimilarities of nucleic or amino acid sequences are therefore used to approximate the relatedness of genomic regions. The *Molecular Clock Hypothesis*, which was first described in the early 1960s, states that genes evolve with a constant mutation rate [79]. In this case, for a constant substitution rate r the genetic distance d of two genes x and y would be given by $d(x, y) = 2r\tau(\mathrm{lca}_T(x, y))$. It can easily be seen that such distances are still ultrametric. The theory has been shown to be a remarkably good approximation for many gene families and it is a key assumption for numerous bioinformatics algorithms and tools [e.g. 9, 52].

However, the *Molecular Clock Hypothesis* fails in the presence of the well-known phenomenon of asymmetric divergence after gene duplication. The first attempt to explain unequal mutation rates of two paralogs after their separation was *Ohno's neofunctionalization* model [60]. He suggested that, after the duplication, one copy will maintain its original function, while the other becomes relieved from negative selective pressure and therefore accumulates mutations. In many cases, the redundant copy would get lost again or pseudogenized, but in other cases, the mutations might lead to the acquisition of new functions that can be completely different from the original one. An example supporting this theory is the evolution of an antifreeze protein from a duplicated sialic acid synthase gene in an Antarctic zoarcid fish enabling it to survive in colder environments than related species [**?**].

Another very prominent model for divergence after duplication is the *Duplication-degeneration-complementation (DDC)* model by Force et al. [25]. There, it is assumed that both copies accumulate mutations in the first phase after duplication as a result of reduced selective pressure ('degeneration phase'). As a consequence, one copy alone can no longer fulfill its original function sufficiently. Rather, they divide the function either on the level of regulation by differential gene expression or on the level of gene products, i.e., the proteins or functional RNAs differ in their functionality. Either way, both copies are needed ('complementation') and must be maintained by selection. The model makes different predictions for the evolution rates after duplication. In the degeneration phase, both copies should show an increased, but approximately equal mutation rate. In the complementation phase, they are again exposed to a negative selective pressure leading to either equal or asymmetric divergence depending on the functions and the specific case.

A closely related theory is the *specialization* model [43, 21]. It proposes that genes can suffer from a so-called adaptive conflict, i.e., the gene is fulfilling two or more functions and is therefore constrained from optimizing either one of them since this would lead to a degeneration of the others. Hence, a duplication might allow the *escape from adaptive conflict (EAC)* and specialization for single functions. As before, the extent of mutation rate asymmetries is case-dependent.

So far, all models predict that at least one of the duplicates acquires new functions. However, there are also known cases in which an increased gene dosage after duplication confers an immediate selective advantage to a species [51, 15]. The original function is therefore maintained by both copies and the (possibly relaxed) selective pressure affects

them both equally.

Numerous other models and extensions to the above-described ones have been proposed, a number of which have been reviewed e.g. by Innan and Kondrashov [44], where the summary in Table 3.1 is taken from.

Model	**Function**			**Evolution rate**	
	Original copy	New copy	Fate-determining mutation	Original copy	New copy
Neofunctionalization	Kept	Novel	Gain-of-function mutations	α	β
DDC	Subfunctionalized	Subfunctionalized	Loss-of-function mutations	β	β'
Specialization (EAC)	Subfunctionalized	Subfunctionalized	Gain-of-function mutations	β	β'
Positive dosage	Kept	Same as original	Not applicable	α'	α'

Table 3.1: Summary of the models of gene-duplication evolution (excerpt and modified from Innan and Kondrashov [44]). The rate α represents the pattern in the pre-duplication phase, where α' indicates a possibly relaxed selective pressure. Accelerated non-synonymous mutation rates (i.e., mutations leading to amino acid substitutions) are indicated by β and β'.

It has to be remarked, that all of these models do not exclude one another, i.e., gene duplication scenarios in biological systems could either be combinations of several models, as e.g. found by Teufel et al. [77], or they could simply coexist. In the latter case, it is likely that all (or at least most) theories have their counterparts in biology.

Anyways, asymmetric divergence rates have been observed in a large number of clades ranging from simple eukaryotes like yeast [48, 14, 68], invertebrates like *Drosophila* [2] to chordates [10, 38, 3, 59]. In order to produce realistic gene history scenarios, it is therefore necessary to take into account unequal divergence after gene duplication.

Divergence After Gene Duplication

To estimate the divergence asymmetry between pairs of paralogs, the sequence divergence to a member of the gene family in a closely related species has to be determined. Commonly used measures of sequence divergence are the rate of synonymous nucleic acid exchanges K_S and the rate of non-synonymous (or amino acid) exchanges K_A [48, 14, 3].

Whilst synonymous mutations do not lead to an amino acid exchange in the resulting protein, this is the case for the latter which are therefore subject to stronger selective pressure. Hence, the ratio $\omega = K_A/K_S$ can be used as a measure for the intensity of selection [14, 44]. In the normal case, non-synonymous mutations are stronger constrained resulting in a ratio $\omega < 1$ (negative selection). In the degeneration scenario, a gene might be relieved from selective pressure ($\omega \approx 1$, neutral selection). On the other hand, a value

of $\omega > 1$ indicates a positive selection, which can, e.g., be the case in the specialization scenario where advantageous mutations for a specific function get enriched.

In a study that examined pairs of ohnologs in yeast species, i.e., paralogous genes that descended from a whole-genome duplication (WGD), the two different measures for rate asymmetry R and R' were introduced [23, 14]. Both measures rely on the amino acid substitution rate K_A, since it better reflects the selection pressure. The R-value is the ratio of the maximal and minimal distances to the outgroup, i.e., $\max(K_a + K_c, K_b + K_c)/\min(K_a + K_c, K_b + K_c)$ for paralog 1 and paralog 2 in Figure 3.9. In contrast, R' is the maximum divided by the minimum of the distances of the terminal branches leading to the respective ohnologs ($\max(K_a, K_b)/\min(K_a, K_b)$). Hence, R' only takes into account the non-shared components (Figure 3.9).

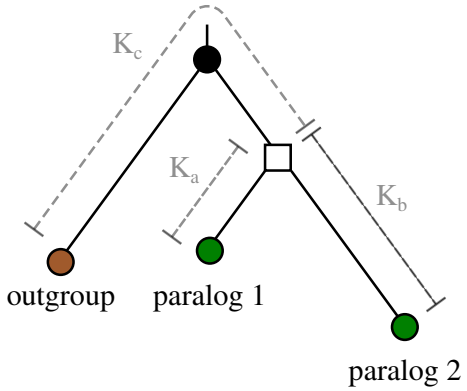

Figure 3.9: Visualization of the quantities necessary to compute the asymmetry measures R and R'. The WGD event is indicated by the square symbol (□). The *outgroup* is branching before the WGD.

Byrne and Wolfe [14] analyzed 653 ohnolog pairs in *Saccharomyces cerevisiae*, *Saccharomyces castellii* and *Candida glabrata* that resulted from whole-genome duplication in a common ancestor of the three species. As outgroups, they used the corresponding orthologs in *Kluyveromyces lactis*, a species that branched before the WGD event.

The resulting R'-values for the 188 ohnolog pairs in *S. cerevisiae* are supplied by Byrne and Wolfe [14]. Therefore, their distribution is used as a reference for modeling realistic substitution rate asymmetries. As by construction $R' > 1$, a Gamma distribution with variable offset on the x-axis was fitted to the observed distribution.

Both the histogram and the fitted Gamma distribution in Figure 3.10 show a concentration at lower asymmetries with a long tail on the right side indicating that large asymmetries like e.g. $R' = 14$ are rare but possible. However, there are a couple of drawbacks arising from the choice of the data set. Firstly, duplicates resulting from a whole-genome duplication may behave differently than small-scale duplicates. It has been observed that the former are on average less divergent in sequence and functionality [35]. The main reason being discussed for this is the effect of dosage advantages and disadvantages. In WGD, the genes of all components of protein complexes are copied resulting in the same relative amounts. In contrast, the imbalance of dosage after duplication of a single component is often deleterious. Whole-genome duplicates have been found to be enriched with such complex-building proteins which is in good accordance with this theory [63]. The higher selective pressure resulting from the maintenance of the interaction may thus be an explanation for the observations.

On the other hand, whole-genome duplicates all emerged at the same point in time,

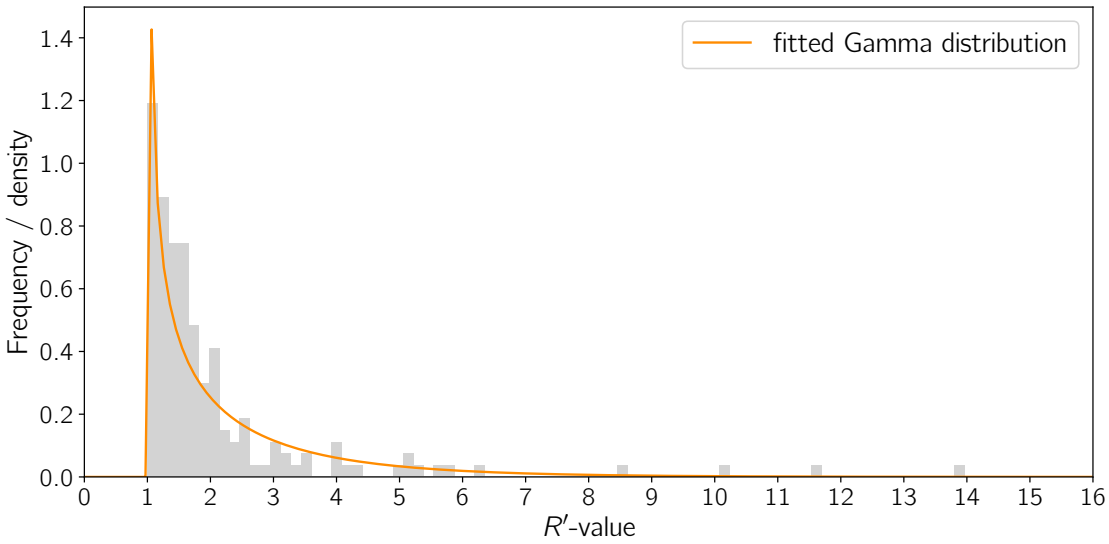

Figure 3.10: Histogram of R'-values measured for whole-genome duplication in *S. cerevisiae* taken from Byrne and Wolfe [14] and the probability density function of the fitted Gamma distribution (shape $k = 0.538$, scale $\theta = 2.183$, offset on x-axis $= 1.007$). The R'-values are grouped into 80 equal-sized bins.

which is not true for small-scale duplicates and the independent development in separate branches of the species tree. This is especially problematic if the widely assumed theory holds that gene copies evolve fastest directly after the duplication and then slow down again [45, 44]. In this case, the expected value of the R' distribution should be highest directly after the degeneration and fate-determining phase and eventually slowly decrease again.

As a conclusion, data from small-scale duplications or perhaps a mixed data set should be preferred. The rate asymmetries after single-gene duplications have been investigated e.g. in rodents and show a magnitude comparable with the yeast data [18, 26]. However, data is available only for few duplicate pairs making the fitting of any theoretical distribution difficult. For this reason, the fitted Gamma distribution described above will be used here to model substitution rate imbalances. The procedure is explained in detail in Section 3.2.3.

Substitution Rate Variance Among Species

Another important source of substitution rate variance is the heterogeneity among the considered species. Mutation rates are negatively correlated with traits like body size and generation time and positively correlated with metabolic rate. Whilst the connection to generation time is explained intuitively by a slower establishment of mutations in the population, the latter is, e.g., linked to a higher abundance of aerobic respiration by-products and a generally increased DNA synthesis [57, 33, 58].

As this also affects the singleton genes, i.e., genes having no paralogs in the same species, it constitutes an important aspect that violates the *Molecular Clock Hypothesis* and should be taken into account for realistic evolution rate modeling. The goal of this section is, therefore, to assign baseline rates for the conserved genes to all nodes and edges of the species tree.

Kishino et al. [50] simulate the development of the mutation rate ϱ by sampling the rate

of the ending node of an edge uv from a log-normal distribution (Lognormal(μ, σ^2)). To avoid bias towards increasing or decreasing rates, the mean μ of the underlying normal distribution is normalized such that $\mathbf{E}(\varrho_v) = \varrho_u$. Since the expected value of a log-normal distribution is given by

$$\mathbf{E}(X) = e^{\mu + \frac{\sigma^2}{2}} \tag{3.12}$$

the parameter μ must be set to

$$\mu = \ln \varrho_u - \frac{\sigma^2}{2} \tag{3.13}$$

to ensure this property. The divergence time of the edge, i.e., $\tau(u) - \tau(v)$ which is positive by construction, multiplied by a value σ_0^2 is used as variance parameter σ^2. Hence, the overall variance is given by

$$\begin{aligned} \mathbf{Var}(\varrho_v) &= (e^{\sigma^2} - 1)\, e^{2\mu + \sigma^2} \\ &= (e^{\sigma^2} - 1)\, e^{2(\ln \varrho_u - \frac{\sigma^2}{2}) + \sigma^2} \\ &= (e^{\sigma^2} - 1)\, \varrho_u^2 \\ &= (e^{\sigma_0^2 (\tau(u) - \tau(v))} - 1)\, \varrho_u^2. \end{aligned} \tag{3.14}$$

Note that this method makes the implicit assumption that the temporal development of the mutation rate is a *geometric Brownian motion* process, i.e., its logarithm follows a *Brownian motion* (also called *Wiener process*). This can easily be seen by the fact that, for a *geometric Brownian motion* process with random variable X and an elapsed time t, the value of X_t is also log-normally distributed [61]. In particular, the expected value and variance are then given by

$$\begin{aligned} \mathbf{E}(X_t) &= X_0 e^{\mu t} \\ \mathbf{Var}(X_t) &= (e^{\sigma^2 t} - 1)\, X_0^2 e^{2\mu t}. \end{aligned} \tag{3.15}$$

By setting $X_0 = \varrho_u$ (the starting rate at vertex u), $\sigma^2 = \sigma_0^2$ (the variable simulation parameter) and $\mu = 0$ (in order to avoid bias towards in- or decreasing values) this becomes

$$\begin{aligned} \mathbf{E}(\varrho_v) &= \varrho_u \\ \mathbf{Var}(\varrho_v) &= (e^{\sigma_0^2 (\tau(u) - \tau(v))} - 1)\, \varrho_u^2 \end{aligned} \tag{3.16}$$

for a specific edge $uv \in E(S)$, which is the same as above.

Having assigned rates to all vertices $v \in V(S)$ in a pre-order traversal, here, the effective rate ϱ_e^0 for an edge $e = uv \in E(S)$ is simply set to the arithmetic mean of its endpoints:

$$\varrho_e^0 = \frac{\varrho_u + \varrho_v}{2}. \tag{3.17}$$

As described in the next section, the rates ϱ_e^0 will be used as factors for the genes that are embedded into the respective edge of the species tree. This introduces a correlation between paralogs in the same species. Furthermore, the *geometric Brownian motion* process generates a correlation between descending branches and their ancestors, referred to as *autocorrelation* [50]. The degree of autocorrelation is controlled by the factor σ_0^2. For an edge $uv \in E(S)$, a high value leads to a large variation of ϱ_v w.r.t. the starting rate ϱ_u and hence results in low autocorrelation. On the other hand, a low value of σ_0^2 indicates strong autocorrelation.

Implementation

As seen in the previous sections, several aspects have to be taken into account to simulate realistic substitution rate imbalances. Therefore, a hierarchical model is used that incorporates both the baseline rates ϱ_e^0 assigned to the branches of the species tree S and the gene-specific effects resulting from unequal fates of gene copies after duplication and horizontal gene transfer events.

The values ϱ_e^0 for the edges $e = uv \in E(S)$ are determined by a pre-order traversal of S as described in Section 3.2.3, where a user-defined mean substitution rate for the conserved members of the gene family is used as starting value ϱ_{ρ_S} or ϱ_{0_S} if the tree is planted, respectively.

For the contribution of asymmetric divergence of paralogs in an extended gene tree T, the following three types of duplication events are defined. Their relative occurrence is controlled by weights that can again be specified by the user:

i) *Conservation*
The child branches of such a duplication event will both be marked as `conserved` (cf. positive dosage model).

ii) *Subfunctionalization*
The child branches of such a duplication event will both be marked as `divergent` [cf. DDC model by 25].

iii) *Neofunctionalization*
One child branch will be marked as `conserved`, the other one as `divergent` [cf. neofunctionalization model by 60].

In the initialization phase, all vertices $u \in V(T)$ are sorted in natural temporal order, i.e., a node v_1 appears before v_2 in the list if $\tau(v_1) > \tau(v_2)$, where ties are broken arbitrarily. Furthermore, it is necessary to keep track of the current number of paralogous genes in each branch of the species tree during the simulation. As mentioned above, the edges of the gene tree T can be marked as either `conserved` or `divergent` depending on the fate of the branch after a duplication event.

In order to record the gene-specific substitution rates ϱ, for each edge $e = uv \in E(T)$ an empty list $\mathfrak{L}_e$ of ordered pairs of the form (τ, ϱ) is initialized, where ϱ is a substitution rate and τ the corresponding time point at which the rate becomes valid during the existence of e. This especially enables the resetting of the gene status to `conserved` in case it is the last survivor in a given species. At the moment, this is the main purpose of the introduction of the lists $\mathfrak{L}_e$. However, future extensions of the model can easily be adapted to consider other events that change the substitution rate during the existence of an edge. Moreover, temporarily restricted rate changes, as suggested by some duplication models [cf. 44], are possible. For easier notation, the i^{th} ordered pair (τ_i, ϱ_i) in $\mathfrak{L}_e$ is denoted by $\mathfrak{L}_{e,i}$. Furthermore, $\tau(\mathfrak{L}_{e,i}) \coloneqq \tau_i$ and $\varrho(\mathfrak{L}_{e,i}) \coloneqq \varrho_i$.

Recall that $0_T\rho_T$ is the first (planted) edge in the gene tree T. The simulation is started by marking $0_T\rho_T$ as `conserved` and appending $(\tau(0_T), 1.0)$ to $\mathfrak{L}_{0_T\rho_T}$. For each vertex u in the sorted list, it then proceeds as follows:

a) *u is a speciation event*
Mark all edges uv with $v \in \mathsf{child}(u)$ the same as $\mathsf{par}(u)u$. Append to $\mathfrak{L}_{uv}$the pair $(\tau(u), \varrho)$ with $\varrho = 1.0$ (uv is `conserved`) or ϱ Gamma-distributed (uv is `divergent`), respectively.

b) *u is a duplication event*
If the edge $\mathsf{par}(u)u$ is marked as `divergent`, then all edges uv with $v \in \mathsf{child}(u)$ are

also marked as `divergent` and corresponding pairs $(\tau(u), \varrho)$ are appended to $\mathfrak{L}_{uv}$, where the values of ϱ are drawn i.i.d. from the Gamma distribution. If $\mathsf{par}(u)u$ is marked as `conserved`, choose between

i) conservation
mark both incident edges below u as `conserved`,

ii) subfunctionalization
mark both incident edges below u as `divergent`, and

iii) neofunctionalization
mark one edge as `conserved` and the other as `divergent`

with the specified weights. Append to $\mathfrak{L}_{uv}$ the pair $(\tau(u), \varrho)$ with $\varrho = 1.0$ (uv is `conserved`) or ϱ Gamma-distributed (uv is `divergent`), respectively.

c) *u is a loss event*
If a single copy is left in the respective species after the loss: Let e^* be the corresponding edge of the remaining copy at $\tau(u)$. Mark e^* as `conserved` and append the pair $(\tau(u), 1.0)$ to $\mathfrak{L}_{e^*}$.

d) *u is an HGT event*
Let v_1 be the copy that remains in the species and v_2 the transferred copy. Mark uv_1 the same as $\mathsf{par}(u)u$ and append $(\tau(u), \varrho)$ to $\mathfrak{L}_{uv_1}$ where ϱ is the last rate that was appended to $\mathfrak{L}_{\mathsf{par}(u)u}$. Mark uv_2 as `divergent` and append $(\tau(u), \varrho)$ to $\mathfrak{L}_{uv_2}$ with ϱ Gamma-distributed.

The simulation ends with the non-loss leaves of the gene. They belong to neither of the categories above, and, therefore, no further actions need to be performed. As a last step, for each edge $e = uv \in E(T)$, the list $\mathfrak{L}_e$ is finalized by appending $(\tau(v), \varrho)$ where ϱ is the last rate that was appended to $\mathfrak{L}_e$ so far. This allows computing the edge length $\ell(e)$ for each edge e in T as a time-weighted mean of the assigned rates:

$$\ell(e) = \varrho_f^0 \sum_{i=1}^{|\mathfrak{L}_e|-1} \varrho(\mathfrak{L}_{e,i}) \cdot (\tau(\mathfrak{L}_{e,i}) - \tau(\mathfrak{L}_{e,i+1})) \tag{3.18}$$

where f is the edge in the species tree S into which e is embedded. The resulting distance function $\ell : E(T) \to \mathbb{R}^+$ induces an additive metric on the set of leaves $L(T)$ (see Section 3.1.3).

The following standard parameterization is used as default:

Description		Default value
mean substitution rate of the conserved genes		1.0
variance σ_0^2 for the baseline substitution rates in S		0.2
Gamma distribution for the (gene-specific) substitution rates > 1 of divergent genes	shape k	0.5
	scale θ	2.2
	offset on the x-axis	1.0
Duplication type weights	conservation	1/3
	subfunctionalization	1/3
	neofunctionalization	1/3

Table 3.2: Default parameterization for the gene tree imbalancing used for the simulations. The parameters for the Gamma distribution were fitted to a yeast data set from Byrne and Wolfe [14], the rest is chosen arbitrarily.

To have the possibility to completely disable the asymmetric divergence after gene duplication, the *conservation* mode was introduced. Among the models (see Table 3.1), this scenario is best explained by those that assume an advantageous dosage effect, in which the function is kept for both copies and the selective pressure is the same for both (possibly slightly relaxed).

The question which of the two theoretic models neofunctionalization and subfunctionalization occurs more often in biologic systems has been addressed in several studies. Therein, the comparison of gene expression profiles in different tissues was used as a proxy to decide for an individual pair of paralogs which of the scenarios is more likely [55, 7]. In this framework, a partitioning of expression regulation indicates subfunctionalization, and, e.g., the expression in a new tissue w.r.t. a related species that branched before the duplication event is characteristic for neofunctionalization. Whilst Lien et al. [55] found that neofunctionalization is most abundant in teleosts, Braasch et al. [7] found more evidence for a dominance of subfunctionalization (also in fish expression data). However, a reanalysis of the data by Sandve et al. [67] showed that these discrepancies are a result of the different methods and interpretation in the two studies. As it is hard to come by reliable numbers that can be used as weights for the choice between duplication event types, here, an *a priori* uniform distribution is assumed for the three modes *conservation*, *subfunctionalization* and *neofunctionalization*. The weights $w_C = w_S = w_N = 1/3$ are thus used as the default setting.

Another possible approach to handle this issue would be to use a prior distribution for the weighting and choose them randomly for an individual gene tree. The same could be applied for parameters like the mean substitution rate of the conserved members of the gene family and the variance factor σ_0^2. The current implementation could easily be extended with such priors in future refinements.

3.2.4 Simulated Measurement Noise

A simulated gene tree T with weight function $\ell : E(T) \to \mathbb{R}^+$, as described so far, induces an additive metric on the set of its leaves. For evaluating the performance of phylogenetic reconstruction algorithms, the distances between the non-loss leaves, i.e., the extant genes at time $\tau = 0$, are of special interest, since they represent the measurable part of the metric in real data.

The number of mutation events that occurred on the path between two genes is not directly observable, in particular, due to backmutations. As a consequence, simple dissimilarity measures like the Hamming distance in pairwise alignments are not additive. This issue can partly be overcome by applying a distance correction with a suitable model for DNA evolution [46, 49, 76]. Such transformations yield good approximations to the true evolutionary distances in most cases, but all models make assumptions for the sequence evolution that might not be realistic. Moreover, they must rely on the quality of the alignment. Hence, both steps can introduce noise and bias into the data.

Another important aspect, that was already mentioned, is that both nucleic acid and protein sequences have a limited length. As the approximation of distances relies on the quantification of discrete evolutionary events, they are difficult to estimate for very short sequences. Also, the DNA evolution models often produce poor results in case the alignment is saturated, i.e., most base or amino acid positions have undergone mutations and backmutations [64].

As a consequence, it is necessary to introduce measurement noise to produce realistic test data for any phylogenetic and orthology detection tool. Two approaches for noise simulation have been implemented: random perturbations drawn from a normal distribution and a systematic disturbance with a wrong tree topology.

Random Perturbations

The first strategy for the introduction of measurement noise aims to avoid a systematical bias. Therefore, random perturbations ϵ_{xy} are drawn from a normal distribution with mean 1 and a user-defined standard deviation σ. The noise is then incorporated into the distance matrix $\mathbf{D}$ (on the leaf set L of a gene tree T) by substituting $\mathbf{D}'(x, y) = \mathbf{D}'(y, x) = \varepsilon_{xy}\mathbf{D}(x, y)$. Note that using multiplicative noise with mean 1 avoids that small distances 'suffer' more from bigger (absolute) perturbations than large distances, as it would be the case for additive noise with mean 0.

Most phylogenetic inference methods rely on additive metric data [cf. 65]. In general, the introduction of noise does not preserve this property, especially by violating the triangle inequality. Whilst additivity is hard to be guaranteed, Algorithm 3 is applied to ensure that the disturbed matrix $\mathbf{D}'$ is still a metric.

The idea of this algorithm is to simply reject all perturbations that violate the triangle inequality. The process is repeated until $\binom{|L|}{2}$ changes have been accepted to ensure an appropriate scaling with the size of the input matrix. Note that not all cell entries will be changed in most cases, whereas some are changed multiple times. In every iteration step, $\mathcal{O}(|L|)$ inequalities have to be checked. For moderate noise, the number of iterations should not exceed $\mathcal{O}(|L|^2)$. Hence, the time complexity of the procedure is at least $\mathcal{O}(|L|^3)$. This approach will be referred to as *Rejection method* in the following.

In an alternative strategy, noise is first added to the distance of every unordered pair $x \neq y \in L$. In a subsequent step, the disturbed matrix is then modified such that it is again a metric. This task is known as the *metric repair* or *metric nearness problem* for which several algorithms have been suggested [e.g. cf. 8, 30]. To this end, random perturbations drawn from a normal distribution with mean 1 and standard deviation σ

Algorithm 3: Introduction of random measurement noise into $\mathbf{D}$

Input: Additive distance matrix $\mathbf{D}$ with dimension $|L| \times |L|$, standard deviation σ
Output: Disturbed matrix $\mathbf{D}'$
Initialize $\mathbf{D}' = \mathbf{D}$.
Initialize $n_{\text{success}} = 0$.
while $n_{success} < \binom{|L|}{2}$ **do**
    Draw $x \neq y \in L$ randomly.
    Draw ε_{xy} from $\mathcal{N}(1, \sigma^2)$.
    Set $d_{\text{old}} = \mathbf{D}'(x, y)$.
    Set $\mathbf{D}'(x, y) = \mathbf{D}'(y, x) \leftarrow \varepsilon_{xy}\mathbf{D}'(x, y)$.
    if $\mathbf{D}'$ *is a metric*
     then
        $n_{\text{success}} \leftarrow n_{\text{success}} + 1$
    else
        Set $\mathbf{D}'(x, y) = \mathbf{D}'(y, x) \leftarrow d_{\text{old}}$.
    end
end

are multiplied to each of the $\binom{|L|}{2}$ distinct distances between the leaves $x \neq y \in L$. In the case of negative values, a new disturbance is drawn. In a subsequent step, one of the algorithms for *sparse metric repair* described by Gilbert and Jain [30] is applied to extract a distance matrix from the disturbed data that again satisfies the properties of a metric.

The first metric repair algorithm that was implemented here is the *Decrease Only Metric Repair (DOMR)* in which a perturbation matrix $\mathbf{P}$ is searched such that $\hat{\mathbf{D}} = \mathbf{D}' + \mathbf{P}$ is a metric and the values in $\mathbf{D}'$ are only decreased. A modified Floyd-Warshall algorithm is used to ensure that $\hat{\mathbf{D}}$ is a metric and that $\mathbf{P}$ is minimal w.r.t. the ℓ_1 norm. The second algorithm is a heuristic for a *General Metric Repair (GMR)*, i.e., there are no restrictions for the matrix $\mathbf{P}$. In contrast to *DOMR*, this approach does not guarantee the resulting matrix to be a metric.

The *Rejection method* seems to produce the most stable results and is therefore used as default.

Wrong Topology Bias

The correct topology of a phylogenetic tree can be inferred from additive metric distance data with the exception of the location of the root [4, 66]. It is therefore of interest to examine the sensitivity to systematic disturbances for tools that explicitly make use of topology reconstruction.

To this end, small contributions of a second distance matrix $\mathbf{D}^*$ of equal dimension are added to $\mathbf{D}$ by forming a convex linear combination:

$$\mathbf{D}' = (1 - \alpha)\mathbf{D} + \alpha\mathbf{D}^* \qquad 0 \leq \alpha \leq 1. \tag{3.19}$$

If both $\mathbf{D}$ and $\mathbf{D}^*$ are metrics, the resulting matrix $\mathbf{D}'$ will also be a metric. The first three properties of metrics are trivially preserved. It can also easily be seen that no triangle is destroyed by considering that multiplying inequalities with a factor greater than zero and adding them preserves the inequality.

For a specific scenario with species tree S and an observable gene tree T giving rise to the distance matrix $\mathbf{D}$, the disturbance matrix $\mathbf{D}^*$ is constructed as follows: A second

gene tree T_2 is simulated along S and a corresponding distance matrix $\mathbf{D}_2$ is computed. If $|L(T)| \leq |L(T_2)|$, the upper left corner of $\mathbf{D}_2$ is simply used as $\mathbf{D}^*$. Otherwise, $\mathbf{D}_2$ is recycled by placing it in multiple rows and columns until the dimension of $\mathbf{D}$ is filled where overhanging cells are cut off.

References

[1] Aho, A. V., Sagiv, Y., Szymanski, T. G., and Ullman, J. D. Inferring a Tree from Lowest Common Ancestors with an Application to the Optimization of Relational Expressions. *SIAM Journal on Computing*, 10(3):405–421, August 1981. ISSN 0097-5397, 1095-7111. doi: 10.1137/0210030.

[2] Assis, R. and Bachtrog, D. Neofunctionalization of young duplicate genes in Drosophila. *Proceedings of the National Academy of Sciences*, 110(43):17409–17414, October 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1313759110.

[3] Assis, R. and Bachtrog, D. Rapid divergence and diversification of mammalian duplicate gene functions. *BMC Evolutionary Biology*, 15(1), December 2015. ISSN 1471-2148. doi: 10.1186/s12862-015-0426-x.

[4] Bandelt, H.-J. and Dress, A. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*, 7(3):309–343, September 1986. ISSN 01968858. doi: 10.1016/0196-8858(86)90038-2.

[5] Böcker, S. and Dress, A. W. Recovering Symbolically Dated, Rooted Trees from Symbolic Ultrametrics. *Advances in Mathematics*, 138(1):105–125, September 1998. ISSN 00018708. doi: 10.1006/aima.1998.1743.

[6] Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M., and Yuan, Y. Predicting function: From genes to genomes and back 1 1Edited by P. E. Wright. *Journal of Molecular Biology*, 283(4):707–725, November 1998. ISSN 00222836. doi: 10.1006/jmbi.1998.2144.

[7] Braasch, I., Gehrke, A. R., Smith, J. J., Kawasaki, K., Manousaki, T., Pasquier, J., Amores, A., Desvignes, T., Batzel, P., Catchen, J., Berlin, A. M., Campbell, M. S., Barrell, D., Martin, K. J., Mulley, J. F., Ravi, V., Lee, A. P., Nakamura, T., Chalopin, D., Fan, S., Wcisel, D., Cañestro, C., Sydes, J., Beaudry, F. E. G., Sun, Y., Hertel, J., Beam, M. J., Fasold, M., Ishiyama, M., Johnson, J., Kehr, S., Lara, M., Letaw, J. H., Litman, G. W., Litman, R. T., Mikami, M., Ota, T., Saha, N. R., Williams, L., Stadler, P. F., Wang, H., Taylor, J. S., Fontenot, Q., Ferrara, A., Searle, S. M. J., Aken, B., Yandell, M., Schneider, I., Yoder, J. A., Volff, J.-N., Meyer, A., Amemiya, C. T., Venkatesh, B., Holland, P. W. H., Guiguen, Y., Bobe, J., Shubin, N. H., Di Palma, F., Alföldi, J., Lindblad-Toh, K., and Postlethwait, J. H. The spotted gar genome illuminates vertebrate evolution and facilitates human-teleost comparisons. *Nature Genetics*, 48(4):427–437, April 2016. ISSN 1546-1718. doi: 10.1038/ng.3526.

[8] Brickell, J., Dhillon, I. S., Sra, S., and Tropp, J. A. The Metric Nearness Problem. *SIAM Journal on Matrix Analysis and Applications*, 30(1):375–396, January 2008. ISSN 0895-4798, 1095-7162. doi: 10.1137/060653391.

[9] Bromham, L. and Penny, D. The modern molecular clock. *Nature Reviews Genetics*, 4(3):216–224, March 2003. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg1020.

[10] Brunet, F. G., Crollius, H. R., Paris, M., Aury, J.-M., Gibert, P., Jaillon, O., Laudet, V., and Robinson-Rechavi, M. Gene Loss and Evolutionary Rates Following Whole-Genome Duplication in Teleost Fishes. *Molecular Biology and Evolution*, 23(9):1808–1816, September 2006. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/msl049.

[11] Buneman, P. A note on the metric properties of trees. *Journal of Combinatorial Theory, Series B*, 17(1):48–50, August 1974. ISSN 00958956. doi: 10.1016/0095-8956(74)90047-1.

[12] Bustin, R. and Messer, H. An equivalent Markov model for Gillespie's Stochastic Simulation Algorithm for biochemical systems. *14th European Signal Processing Conference*, September 2006.

[13] Byrka, J., Guillemot, S., and Jansson, J. New results on optimizing rooted triplets consistency. *Discrete Applied Mathematics*, 158(11):1136–1147, June 2010. ISSN 0166218X. doi: 10.1016/j.dam.2010.03.004.

[14] Byrne, K. P. and Wolfe, K. H. Consistent Patterns of Rate Asymmetry and Gene Loss Indicate Widespread Neofunctionalization of Yeast Genes After Whole-Genome Duplication. *Genetics*, 175 (3):1341–1350, March 2007. ISSN 0016-6731, 1943-2631. doi: 10.1534/genetics.106.066951.

[15] Conant, G. C. and Wolfe, K. H. Increased glycolytic flux as an outcome of whole-genome duplication in yeast. *Molecular Systems Biology*, 3:129, 2007. ISSN 1744-4292. doi: 10.1038/msb4100170.

[16] Corneil, D. G., Perl, Y., and Stewart, L. K. A Linear Recognition Algorithm for Cographs. *SIAM Journal on Computing*, 14(4):926–934, November 1985. ISSN 0097-5397, 1095-7111. doi: 10.1137/0214065.

[17] Crespelle, C. Linear-Time Minimal Cograph Editing. 2019.

[18] Cusack, B. P. and Wolfe, K. H. Not Born Equal: Increased Rate Asymmetry in Relocated and Retrotransposed Rodent Gene Duplicates. *Molecular Biology and Evolution*, 24(3):679–686, December 2006. ISSN 0737-4038, 1537-1719. doi: 10.1093/molbev/msl199.

[19] Dalquen, D. A., Anisimova, M., Gonnet, G. H., and Dessimoz, C. ALF—A Simulation Framework for Genome Evolution. *Molecular Biology and Evolution*, 29(4):1115–1123, April 2012. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/msr268.

[20] Deng, Y. and Fernández-Baca, D. Fast Compatibility Testing for Rooted Phylogenetic Trees. page 12 pages, 2016. doi: 10.4230/LIPICS.CPM.2016.12.

[21] Des Marais, D. L. and Rausher, M. D. Escape from adaptive conflict after duplication in an anthocyanin pathway gene. *Nature*, 454(7205):762–765, August 2008. ISSN 1476-4687. doi: 10.1038/nature07092.

[22] Diestel, R. *Graph Theory*. Springer Berlin Heidelberg, New York, NY, 2017. ISBN 978-3-662-53621-6.

[23] Fares, M. A., Byrne, K. P., and Wolfe, K. H. Rate Asymmetry After Genome Duplication Causes Substantial Long-Branch Attraction Artifacts in the Phylogeny of Saccharomyces Species. *Molecular Biology and Evolution*, 23(2):245–253, February 2006. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/msj027.

[24] Fitch, W. M. Homology. *Trends in Genetics*, 16(5):227–231, May 2000. ISSN 01689525. doi: 10.1016/S0168-9525(00)02005-9.

[25] Force, A., Lynch, M., Pickett, F. B., Amores, A., Yan, Y. L., and Postlethwait, J. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151(4):1531–1545, April 1999. ISSN 0016-6731.

[26] Gayral, P., Caminade, P., Boursot, P., and Galtier, N. The evolutionary fate of recently duplicated retrogenes in mice. *Journal of Evolutionary Biology*, 20(2):617–626, March 2007. ISSN 1010-061X, 1420-9101. doi: 10.1111/j.1420-9101.2006.01245.x.

[27] Geiß, M., Chávez, E., González Laffitte, M., López Sánchez, A., Stadler, B. M. R., Valdivia, D. I., Hellmuth, M., Hernández Rosales, M., and Stadler, P. F. Best match graphs. *Journal of Mathematical Biology*, 78(7):2015–2057, June 2019. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-019-01332-9.

[28] Geiß, M., Laffitte, M. E. G., Sánchez, A. L., Valdivia, D. I., Hellmuth, M., Rosales, M. H., and Stadler, P. F. Best match graphs and reconciliation of gene trees with species trees. *Journal of Mathematical Biology*, January 2020. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-020-01469-y.

[29] Geiß, M., Stadler, P. F., and Hellmuth, M. Reciprocal best match graphs. *Journal of Mathematical Biology*, 80(3):865–953, February 2020. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-019-01444-2.

[30] Gilbert, A. C. and Jain, L. If it ain't broke, don't fix it: Sparse metric repair. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 612–619, Monticello, IL, USA, October 2017. IEEE. ISBN 978-1-5386-3266-6. doi: 10.1109/ALLERTON.2017.8262793.

[31] Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, December 1976. ISSN 00219991. doi: 10.1016/0021-9991(76)90041-3.

[32] Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, December 1977. ISSN 0022-3654, 1541-5740. doi: 10.1021/j100540a008.

[33] Gillooly, J. F., Allen, A. P., West, G. B., and Brown, J. H. The rate of DNA evolution: Effects of body size and temperature on the molecular clock. *Proceedings of the National Academy of Sciences*, 102(1):140–145, January 2005. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0407735101.

[34] Gordon, A. D. A Review of Hierarchical Classification. *Journal of the Royal Statistical Society. Series A (General)*, 150(2):119, 1987. ISSN 00359238. doi: 10.2307/2981629.

[35] Hakes, L., Pinney, J. W., Lovell, S. C., Oliver, S. G., and Robertson, D. L. All duplicates are not equal: The difference between small-scale and genome duplication. *Genome Biology*, 8(10):R209, 2007. ISSN 1465-6906. doi: 10.1186/gb-2007-8-10-r209.

[36] Hellmuth, M. and Wieseke, N. From Sequence Data Including Orthologs, Paralogs, and Xenologs to Gene and Species Trees. In Pontarotti, P., editor, *Evolutionary Biology*, pages 373–392. Springer International Publishing, Cham, 2016. ISBN 978-3-319-41323-5 978-3-319-41324-2. doi: 10.1007/978-3-319-41324-2_21.

[37] Hellmuth, M., Wieseke, N., Lechner, M., Lenhof, H.-P., Middendorf, M., and Stadler, P. F. Phylogenomics with paralogs. *Proceedings of the National Academy of Sciences*, 112(7):2058–2063, February 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1412770112.

[38] Hellsten, U., Khokha, M. K., Grammer, T. C., Harland, R. M., Richardson, P., and Rokhsar, D. S. Accelerated gene evolution and subfunctionalization in the pseudotetraploid frog Xenopus laevis. *BMC Biology*, 5(1):31, 2007. ISSN 17417007. doi: 10.1186/1741-7007-5-31.

[39] Henzinger, M. R. and King, V. Randomized dynamic graph algorithms with polylogarithmic time per operation. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing - STOC '95*, pages 519–527, Las Vegas, Nevada, United States, 1995. ACM Press. ISBN 978-0-89791-718-6. doi: 10.1145/225058.225269.

[40] Hernandez-Rosales, M., Hellmuth, M., Wieseke, N., Huber, K. T., Moulton, V., and Stadler, P. F. From event-labeled gene trees to species trees. *BMC Bioinformatics*, 13(S19), December 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-S19-S6.

[41] Hernandez-Rosales, M., Wieseke, N., Hellmuth, M., and Stadler, P. F. Simulation of gene family histories. *BMC Bioinformatics*, 15(S3), February 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-S3-A8.

[42] Holm, J., de Lichtenberg, K., and Thorup, M. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48(4):723–760, July 2001. ISSN 00045411. doi: 10.1145/502090.502095.

[43] Hughes, A. L. The evolution of functionally novel proteins after gene duplication. *Proceedings. Biological Sciences*, 256(1346):119–124, May 1994. ISSN 0962-8452. doi: 10.1098/rspb.1994.0058.

[44] Innan, H. and Kondrashov, F. The evolution of gene duplications: Classifying and distinguishing between models. *Nature Reviews Genetics*, 11(2):97–108, February 2010. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg2689.

[45] Jordan, I. K., Wolf, Y. I., and Koonin, E. V. Duplicated genes evolve slower than singletons despite the initial rate increase. *BMC Evolutionary Biology*, page 11, 2004.

[46] Jukes, T. H. and Cantor, C. R. Evolution of Protein Molecules. In *Mammalian Protein Metabolism*, pages 21–132. Elsevier, 1969. ISBN 978-1-4832-3211-9. doi: 10.1016/B978-1-4832-3211-9.50009-7.

[47] Keller-Schmidt, S. and Klemm, K. A model of macroevolution as a branching process based on innovations. *Advances in Complex Systems*, 15(07):1250043, October 2012. ISSN 0219-5259, 1793-6802. doi: 10.1142/S0219525912500439.

[48] Kellis, M., Birren, B. W., and Lander, E. S. Proof and evolutionary analysis of ancient genome duplication in the yeast Saccharomyces cerevisiae. *Nature*, 428(6983):617–624, April 2004. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature02424.

[49] Kimura, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120, December 1980. ISSN 0022-2844.

[50] Kishino, H., Thorne, J. L., and Bruno, W. J. Performance of a Divergence Time Estimation Method under a Probabilistic Model of Rate Evolution. *Molecular Biology and Evolution*, 18(3):352–361, March 2001. ISSN 1537-1719, 0737-4038. doi: 10.1093/oxfordjournals.molbev.a003811.

[51] Kondrashov, F. A., Rogozin, I. B., Wolf, Y. I., and Koonin, E. V. Selection in the evolution of gene duplications. *Genome Biology*, 3(2):RESEARCH0008, 2002. ISSN 1474-760X. doi: 10.1186/gb-2002-3-2-research0008.

[52] Kumar, S. Molecular clocks: Four decades of evolution. *Nature Reviews Genetics*, 6(8):654–662, August 2005. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg1659.

[53] Lafond, M., Meghdari Miardan, M., and Sankoff, D. Accurate prediction of orthologs in the presence of divergence after duplication. *Bioinformatics*, 34(13):i366–i375, July 2018. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/bty242.

[54] Lechner, M., Findeiß, S., Steiner, L., Marz, M., Stadler, P. F., and Prohaska, S. J. Proteinortho: Detection of (Co-)orthologs in large-scale analysis. *BMC Bioinformatics*, 12(1), December 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-124.

[55] Lien, S., Koop, B. F., Sandve, S. R., Miller, J. R., Kent, M. P., Nome, T., Hvidsten, T. R., Leong, J. S., Minkley, D. R., Zimin, A., Grammes, F., Grove, H., Gjuvsland, A., Walenz, B., Hermansen, R. A., von Schalburg, K., Rondeau, E. B., Di Genova, A., Samy, J. K. A., Olav Vik, J., Vigeland, M. D., Caler, L., Grimholt, U., Jentoft, S., Våge, D. I., de Jong, P., Moen, T., Baranski, M., Palti, Y., Smith, D. R., Yorke, J. A., Nederbragt, A. J., Tooming-Klunderud, A., Jakobsen, K. S., Jiang, X., Fan, D., Hu, Y., Liberles, D. A., Vidal, R., Iturra, P., Jones, S. J. M., Jonassen, I., Maass, A., Omholt, S. W., and Davidson, W. S. The Atlantic salmon genome provides insights into rediploidization. *Nature*, 533 (7602):200–205, December 2016. ISSN 1476-4687. doi: 10.1038/nature17164.

[56] López Sánchez, A. *Estudio Computacional de Escenarios Evolutivos (Computational Study of Evolutionary Scenarios)*. Bachelor's Thesis, Universidad Nacional Autónoma de México (UNAM), Juriquilla, Querétaro, March 2019.

[57] Martin, A. P. and Palumbi, S. R. Body size, metabolic rate, generation time, and the molecular clock. *Proceedings of the National Academy of Sciences*, 90(9):4087–4091, May 1993. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.90.9.4087.

[58] Nabholz, B., Glemin, S., and Galtier, N. Strong Variations of Mitochondrial Mutation Rate across Mammals–the Longevity Hypothesis. *Molecular Biology and Evolution*, 25(1):120–130, November 2007. ISSN 0737-4038, 1537-1719. doi: 10.1093/molbev/msm248.

[59] Nembaware, V. Impact of the Presence of Paralogs on Sequence Divergence in a Set of Mouse-Human Orthologs. *Genome Research*, 12(9):1370–1376, September 2002. ISSN 10889051. doi: 10.1101/gr.270902.

[60] Ohno, S. *Evolution by Gene Duplication*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1970. ISBN 978-3-642-86661-6 978-3-642-86659-3. doi: 10.1007/978-3-642-86659-3.

[61] Øksendal, B. K. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer, Berlin ; New York, 5th ed edition, 1998. ISBN 978-3-540-63720-2.

[62] Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D., and Maltsev, N. The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences*, 96(6):2896–2901, March 1999. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.96.6.2896.

[63] Papp, B., Pál, C., and Hurst, L. D. Dosage sensitivity and the evolution of gene families in yeast. *Nature*, 424(6945):194–197, July 2003. ISSN 1476-4687. doi: 10.1038/nature01771.

[64] Philippe, H., Brinkmann, H., Lavrov, D. V., Littlewood, D. T. J., Manuel, M., Wörheide, G., and Baurain, D. Resolving Difficult Phylogenetic Questions: Why More Sequences Are Not Enough. *PLoS Biology*, 9(3):e1000602, March 2011. ISSN 1545-7885. doi: 10.1371/journal.pbio.1000602.

[65] Retzlaff, N. and Stadler, P. F. Phylogenetics beyond biology. *Theory in Biosciences = Theorie in Den Biowissenschaften*, 137(2):133–143, November 2018. ISSN 1611-7530. doi: 10.1007/s12064-018-0264-7.

[66] Saitou, N. and Nei, M. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, July 1987. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040454.

[67] Sandve, S. R., Rohlfs, R. V., and Hvidsten, T. R. Subfunctionalization versus neofunctionalization after whole-genome duplication. *Nature Genetics*, 50(7):908–909, July 2018. ISSN 1546-1718. doi: 10.1038/s41588-018-0162-4.

[68] Scannell, D. R. and Wolfe, K. H. A burst of protein sequence evolution and a prolonged period of asymmetric evolution follow gene duplication in yeast. *Genome Research*, 18(1):137–147, November 2007. ISSN 1088-9051. doi: 10.1101/gr.6341207.

[69] Schaller, D. *Inference of Best Matches From Evolutionary Distance Data*. Master's thesis, University of Leipzig, Leipzig, October 2019.

[70] Semple, C. and Steel, M. A. *Phylogenetics*. Number 24 in Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, Oxford ; New York, 2003. ISBN 978-0-19-850942-4.

[71] Simões Pereira, J. A note on the tree realizability of a distance matrix. *Journal of Combinatorial Theory*, 6(3):303–310, April 1969. ISSN 00219800. doi: 10.1016/S0021-9800(69)80092-X.

[72] Simonsen, M., Mailund, T., and Pedersen, C. N. S. Rapid Neighbour-Joining. In Crandall, K. A. and Lagergren, J., editors, *Algorithms in Bioinformatics*, volume 5251, pages 113–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-87360-0 978-3-540-87361-7. doi: 10.1007/978-3-540-87361-7_10.

[73] Spielman, S. J. and Wilke, C. O. Pyvolve: A Flexible Python Module for Simulating Sequences along Phylogenies. *PLOS ONE*, 10(9):e0139047, September 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0139047.

[74] Stadler, P. F., Geiß, M., Schaller, D., Sánchez, A. L., González, M. E., Valdivia, D. I., Hellmuth, M., and Rosales, M. H. From Best Hits to Best Matches. *arXiv:2001.00958 [q-bio]*, January 2020.

[75] Tatusov, R. L. A Genomic Perspective on Protein Families. *Science*, 278(5338):631–637, October 1997. ISSN 00368075, 10959203. doi: 10.1126/science.278.5338.631.

[76] Tavaré, S. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. *Lectures on Mathematics in the Life Sciences*, 17:57–86, 1986.

[77] Teufel, A. I., Liu, L., and Liberles, D. A. Models for gene duplication when dosage balance works as a transition state to subsequent neo-or sub-functionalization. *BMC evolutionary biology*, 16:45, February 2016. ISSN 1471-2148. doi: 10.1186/s12862-016-0616-1.

[78] Wu, B. Y. Constructing the Maximum Consensus Tree from Rooted Triples. *Journal of Combinatorial Optimization*, 8(1):29–39, March 2004. ISSN 1382-6905. doi: 10.1023/B:JOCO.0000021936.04215.68.

[79] Zuckerkandl, E. and Pauling, L. *Molecular Disease, Evolution, and Genic Heterogeneity*. Academic Press, 1962.

Notation

Graphs

$G = (V, E)$ – undirected graph
V – set of vertices
E – set of edges
$\vec{G}$ – directed graph
x, y – vertices
e, f – edges
xy – undirected edge between x and y
(x, y) – directed edge between x and y
P – path
$\mathsf{deg}(x)$ – degree of vertex x
$\mathcal{C}$ – set of connected components

(Phylogenetic) Trees

$T = (V, E)$ – rooted (gene) tree
T_{obs} – observable gene tree
$\overline{T}$ – unrooted (gene) tree
S – species tree
$0_T, 0_S$ – planted root
ρ_T, ρ_S – root (first branching event)
u, v, w – vertices
l, x, y, z – genes / leaves
r, s, s_1, s_2 – species
$\mathsf{par}(v)$ – parent of v
$\mathsf{child}(v)$ – set of children of v
$V^0(T)$ – inner vertices
$L(T)$ – leaves
$L[s]$ – leaves of color/species s
$T(v)$ – subtree rooted at v
$\preceq_T$ – ancestor relation
$\mathrm{lca}_T(A)$ – last common ancestor of set A
$xy|z$ – rooted triple
$(xy|uv)$ – quartet relation
$\sigma(x)$ – leaf coloring map
$\mu(v)$ – reconciliation map
$\tau(v)$ – dating function
$d_\tau(x, y)$ – divergence time
$\ell(e)$ – length of edge e
$d(x, y)$ – (evolutionary) distance

Event labeling

$t(v)$ – event labeling map
$\circledcirc$ – (planted) root
$\odot$ – leaf
● – speciation
□ – duplication
△ – horizontal gene transfer (HGT)
$\dashv/*$ – loss

Orthology and Best Matches

Θ – (true) orthology graph
$\overline{\Theta}$ – (true) paralogy graph
$\vec{G}$ – colored Best Match Graph (cBMG)
G – colored Reciprocal Best Match Graph (cRBMG)
$x \to y$ – best match relation
$\langle a_1 b c a_2 \rangle$ – induced P_4

Inference methods

$\mathbf{D}$ – distance matrix
ϵ – relative tolerance threshold
η – discarding threshold for outgroup species
$\mathcal{R}$ – triple set
$\mathcal{L}$ – leaf set
$\mathcal{H}$ – auxiliary graph (`BUILD`)
L_e – leaf set (extant genes)
L_0 – leaf set (losses)
Z_v – set of outgroups for vertex $v \in V(S)$

Simulation

d, l, h – event rates
$g \in G(\tau)$ – extant gene at time τ
$q \in Q$ – event type
$\xi = (g, q)$ – 'reaction' in Gillespie algorithm
$r_\xi(\tau)$ – rate of reaction ξ
$R(\tau)$ – total rate
$\Delta\tau$ – waiting time
$\mathcal{P}$ – priority queue
K_A – non-synonymous substitution rate
K_S – synonymous substitution rate
ω – measure for selection pressure
R, R' – asymmetry measures
ϱ – evolution rate
k, θ – shape and scale (Gamma distribution)
$\mathfrak{L}_e$ – list of assigned rates to edge e
w_C, w_N, w_S – weights for duplication types
σ – standard deviation (normal distribution)
α – contribution of the disturbance matrix in a convex combination
$\mathbf{D}'$ – disturbed distance matrix