
NOTUNG-DM : Quick Start Guide

Date: January 18, 2016

© Copyright 2015 by the Notung Development Team.

The NOTUNG-DM software package is provided “as is” without warranty of any kind. In no event shall the authors or their employers be held responsible for any damage or inconvenience resulting from the use of this software.

Development of functionality to support the analysis of multidomain evolution was supported in part by National Science Foundation Grant DBI1262593 and Human Frontier Science Program grant RGP0043/2013. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Human Frontier Science Program.

Chapter 1

Introduction

NOTUNG-DM is a program for reconstructing the history of events in the evolution of a multidomain family. Domains, sequence fragments that encode protein modules with a distinct structure and function, are the basic building blocks of proteins. Multidomain protein families evolve via domain shuffling; that is, transfer, insertion, duplication, and deletion of these sequence fragments. This process can result in families of genes that share common ancestry and encode proteins with similar, but not necessarily identical, domain content. The individual domains in such a family do not necessarily have the same history.

In our framework, a multidomain family is modeled as a set of domains co-evolving with a gene tree. The history of each individual domain is represented by a tree, provided by the user and typically constructed from the amino acid sequence of the domain family using standard phylogenetic methods. If a domain architecture¹, once formed, subsequently evolved by vertical descent without further gain or loss, then the gene tree and the individual domain trees will have the same topology. Topological disagreement is evidence of domain shuffling. Based on this observation, NOTUNG-DM infers domain events through a formal process of tree comparison called reconciliation.

NOTUNG-DM is an extension of NOTUNG-2.8, which supports gene tree - species tree reconciliation. NOTUNG-DM can perform all of the same functions for reconciling gene trees with species trees. Only a subset of NOTUNG-2.8's functions can be applied to domain trees. [Table 1.1](#) lists the various functions that can be performed on domain, gene, and species trees in NOTUNG-DM. NOTUNG-2.8 gene tree functions that are also required for analysis of multidomain evolution are described in this manual. Descriptions of functions not included here can be found in the Notung-2.8 Manual. [Table 1.2](#) summarizes both NOTUNG-DM and NOTUNG-2.8 functions and provides references to the relevant documentation for each function.

¹The “domain architecture” of a protein is the set of domains it contains. In some contexts, the term “domain architecture” is used to describe an ordered list of domains. Here, “domain architecture” denotes an unordered set.

Task	Domain Tree	Gene Tree	Species Tree
History	✓	✓	✓
Reconcile	✓	✓	×
Root	×	✓	×
Rearrange	×	✓	×
Resolve	×	✓	×
Annotate	✓	✓	✓

Table 1.1: NOTUNG-DM task modes for each tree type.

Functionality	Domain Tree	Gene Tree	Species Tree
History	Notung-2.8 Manual Chap. 9		
Annotate	Notung-2.8 Manual Chap. 10		
Tree Appearance	Notung-2.8 Manual Sec. 11.2-3		
Save PNG	Notung-2.8 Manual Sec. 3.3		
Reconcile	Chapter 3	Notung-2.8 Manual Chap. 5	×
Save Tree	Section 3.3.1	Notung-2.8 Manual Appendix A.3	
Tree Stats	Section 3.3.3	Notung-2.8 Manual Sec. 3.4	
Summary Files	Section 3.3.4	Notung-2.8 Manual Chap. 5	×
Non-binary Trees	×	Notung-2.8 Manual Sec. 5.3	
Command-line	×	Notung-2.8 Manual Chap. 12	
Root	×	Notung-2.8 Manual Chap. 6	×
Rearrange	×	Notung-2.8 Manual Chap. 7	×
Resolve	×	Notung-2.8 Manual Chap. 8	×
Homology	×	Notung-2.8 Manual Sec. 5.5	×

Table 1.2: Guide to documentation of NOTUNG-DM and NOTUNG-2.8 functions.

1.1 How to use this manual

This manual provides an introduction to NOTUNG-DM, and gives step-by-step instructions for NOTUNG-DM's tasks and visualization features.

[Chapter 2](#) introduces reconciliation and the theoretical issues related to reconstructing multidomain evolution in some detail. [Section 2.1](#) reviews gene tree-species tree reconciliation, as performed in NOTUNG-2.8. [Section 2.2](#) describes our event-based model of multidomain evolution. In [Section 2.2.1](#), we discuss how reconciliation is used in NOTUNG-DM to infer the history of domain events in the context of this model. Finally, [Section 2.2.2](#) compares our event-based reconciliation approach to Wagner parsimony, which has historically been used to infer domain gain and loss. We demonstrate with an example that reconciliation can reconstruct histories with parallel gains or losses, where Wagner parsimony cannot.

Multidomain event inference using the NOTUNG-DM graphical user interface (GUI) is illustrated in [Chapter 3](#) through a series of worked examples. Information about input/output and tree file formats is also provided. These worked examples span the commonly encountered events of domain duplication, loss, insertion, and transfer. A user seeking documentation on running the NOTUNG-DM software can proceed directly to [Chapter 3](#). However, for an in-depth discussion of how reconciliation is used to infer domain event histories and how to interpret the results, the reader is encouraged to read [Chapter 2](#) first.

NOTUNG-DM infers the history of each domain family separately. Users may wish to merge the histories of individual domain families to derive a composite history of the domain architecture as a whole across the gene tree. NOTUNG-DM cannot merge multiple domain histories automatically, but the output from NOTUNG-DM can be used to derive a composite history manually. This process is formalized in detail in [Chapter 4](#). [Section 4.3](#) illustrates the creation of a composite history using the example family encountered in [Chapter 3](#). Pseudocode ([Alg. 4.1](#)) is also provided for this procedure for the technically minded user.

1.2 How to cite NOTUNG-DM and NOTUNG-2.8

If you use NOTUNG-DM or NOTUNG-2.8 in a published analysis, please cite the relevant articles.

NOTUNG-DM's reconciliation algorithms for inferring multidomain evolution is described in:

M. Stolzer, K.M. Siewert, H. Lai, M. Xu, D. Durand. *Event inference in multidomain families with phylogenetic reconciliation*. BMC Bioinformatics, 16(S14):S8, 2015.

NOTUNG-2.8's algorithms for reconciling gene and species trees using an event model that includes both duplication and transfer events are described in:

M. Stolzer, H. Lai, M. Xu, D. Sathaye, B. Vernot, D. Durand. *Inferring Duplications, Losses, Transfers, and Incomplete Lineage Sorting with Non-Binary Species Trees*. Bioinformatics 28: i409-i415, 2012.

NOTUNG-2.8 uses event parsimony to root a gene tree or rearrange weak branches in a gene tree. Algorithms implementing these functions are described in :

D. Durand, B. V. Halldorsson, B. Vernot. *A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction*. Journal of Computational Biology, 13(2):320-335, 2006.

1.3 Downloading and Running NOTUNG-DM

The NOTUNG-DM Package (NOTUNG-DM.zip) can be downloaded from the website <http://www.cs.cmu.edu/~durand/Notung/download.html>. When this file is unzipped, a folder called NOTUNG-DM-BETA will be created that includes this manual, a folder containing the trees used in the worked examples in this manual, the journal article describing domain tree reconciliation algorithm implemented in NOTUNG-DM [19], and the NOTUNG-DM executable file, NOTUNG-DM.jar.

NOTUNG-DM is supported on Windows (versions 7 and later), Mac OSX (versions 10.5 and later), and Linux. To run NOTUNG-DM, Java 1.5 or later must be installed on your computer. NOTUNG-DM has been tested under Java 1.5-6, but should work for newer versions of Java.

To download NOTUNG-DM:

Go to <http://www.cs.cmu.edu/~durand/Notung/download.html>

To unzip NOTUNG-DM.zip:

- *Windows or Mac:* Double click on NOTUNG-DM.zip.
- *Mac or Linux:* Execute the command “`unzip NOTUNG-DM.zip`”.

Because NOTUNG-DM comes as a jar file, there is no need for installation. Simply navigate to the directory containing NOTUNG-DM.jar and launch the program.

To launch NOTUNG-DM:

- *Windows or Mac:* Double click on the file NOTUNG-DM.jar².
- *Mac or Linux:* From the command line, enter:
“`java -jar PATH_TO_NOTUNGDM/NOTUNG-DM.jar`”. If the current working directory contains the NOTUNG-DM executable, then enter:
“`java -jar NOTUNG-DM.jar`”.

²If jar files are not automatically associated with java, you may need to right-click on the file NOTUNG-DM.jar and select “Open with” from the pop-up menu

The graphical user interface was partially constructed using the tree visualization library provided by FORESTER (version 1.92) [21]. Cycle detection, used when inferring transfers, utilizes the jGraph library.

Chapter 2

Background.

The genes that encode a family of multidomain proteins are characterized by a mosaic of sequence fragments called domains. Each domain encodes a structural or functional module. Domains can be duplicated within the same gene or can be copied and inserted into another gene. As a result, different parts of the full gene sequence can have different histories.

Events that shape the evolution of such families include events that affect the gene, i.e., speciation, gene duplication, gene loss, and gene transfer, as well as events that affect the mosaic nature of multidomain families, including acquisition of a new sequence fragment by an existing gene, internal domain duplication, and domain loss. The acquired sequence fragment may be a copy of a domain donated by a different gene within the *same* genome (domain insertion) or donated by a gene in a *different* genome (horizontal domain transfer). We refer to this process, which includes domain duplications, insertions, transfers, and losses, as *domain shuffling*.

NOTUNG-DM reconstructs the evolutionary history of a multidomain family by comparing domain, gene, and species trees, using a general approach called reconciliation. Reconciliation is the process of establishing a correspondence between evolutionary histories of related entities at two levels of organization. A mapping is constructed between nodes in the trees representing the history of each entity. This mapping reveals which ancestral entities coexisted, identifies cases where the evolution of the two entities did not proceed in a coordinated manner on both levels, and infers the events responsible for this discordance. This general approach has been used in several biological contexts: the co-evolution of symbiotic species and their hosts, the evolution of species in the context of geographical change, and the evolution of genes in the context of species evolution. All of these involve two entities at different levels of organization in a relationship that is not symmetric: one entity (e.g., the gene or symbiont) evolves in the context of a second entity (e.g., the species or host).

NOTUNG-DM extends the reconciliation framework to three levels of organization: a domain tree evolving within a gene tree, which in turn is evolving within a species tree. The events in the history of a multidomain family are inferred by reconciling a domain tree with a gene tree that has been previously been reconciled with a species tree. Consideration of the co-evolution of domains with both genes and species enables our algorithm to distinguish between domain losses and gene losses and between domain insertions within the

same genome and domain transfers across genomes. Further, our algorithm can determine whether a domain tree co-divergence is due to a species divergence (i.e., a speciation) or a gene divergence (i.e., gene duplication or transfer).

Domain tree-gene tree-species tree reconciliation is based on gene tree-species tree reconciliation; many of the issues that arise here also arise in multidomain reconciliation. Therefore, we begin with a brief introduction to gene tree-species tree reconciliation, focusing on aspects that are most relevant to domain tree reconciliation.

2.1 Gene tree - species tree reconciliation

Disagreement between gene and species trees is evidence that genes diverged through processes other than speciation, including gene duplication, gene loss and horizontal gene transfer. A gene family without gene gain or gene loss will have the same history as the host species. Topological disagreement between gene and species trees is, therefore, evidence of a history of events other than co-divergence with the species tree. The goal of gene tree-species tree reconciliation is to infer the history of events that best explains this disagreement, with respect to a parsimony-based or probabilistic optimization criterion. Reconciliation algorithms differ with respect to the event model (i.e., the set of possible events that are considered possible causes of disagreement) and the optimization criterion. Here, we consider gene family evolution with three possible events, gene duplication, horizontal gene transfer and gene loss (a *DTL* event model). Gene duplication and gene transfer create a new member of the gene family and a new divergence in the gene tree. Gene loss eliminates a family member and prunes a branch, or subtree, from the gene tree. Here, we focus on a parsimony framework in which we seek the minimum cost event history that explains the topological disagreement. Given a reconciliation with n_D duplications, n_T transfers, and n_L losses, the cost of the reconciliation is the weighted sum

$$C_D n_D + C_T n_T + C_L n_L,$$

where C_D , C_T , and C_L are the duplication, transfer, and loss costs, respectively. Note that although co-divergence with the species tree also creates a new divergence in the gene tree, we do not assign a cost to co-divergence or speciation events.

Reconciliation takes as input a rooted, binary gene tree, a rooted, binary species tree¹, and the association between present-day genes and present-day species, i.e., a mapping between the leaves of the gene tree and the leaves of the species tree. The output is a reconciled gene tree; that is, an augmented gene tree in which

- each ancestral gene is labeled with its associated species;
- each node is labeled with the event that caused the divergence at that node; and

¹Gene tree-species tree reconciliation is also possible with a non-binary gene tree or a non-binary species tree, but not both simultaneously. Currently, multidomain analysis in NOTUNG-DM is restricted to binary trees, so we will not discuss this further here. The use of non-binary trees in NOTUNG-2.8 is described in Notung-2.8 Manual Sec. 5.3.

- additional “loss nodes” are inserted in the gene tree to show the location of inferred losses. Loss nodes are also labeled with the species in which the loss occurred.

For gene families in species in which horizontal gene transfer does not occur, a restricted *Duplication-Loss* (*DL*) model may be used in which gene duplication and gene loss are the only events that can be evoked to explain gene tree disagreement. The choice of event model for analyzing a particular gene depends on whether this family is found in species in which horizontal transfer occurs. Current knowledge suggests that horizontal transfer is common in bacteria and rare in metazoa, with increasing evidence for horizontal transfer in fungi and some single cell eukaryotes. Models that include transfers are more complicated than those do not, making it desirable to avoid models with transfers unless the biological system demands them.

As an example, Fig. 2.1 shows the reconciliation of a gene tree (Fig. 2.1(b)) for a hypothetical gene family with one gene in each of four fungal species in the species tree (Fig. 2.1(a)). The mapping between present-day genes and present-day species is encoded in the leaf labels in the gene tree.

The gene tree topology differs from the species tree topology, indicating that the gene family evolved through events other than co-divergence. The most parsimonious reconciliation under the DL model posits that one gene divergence arose via a gene duplication (shown as a red square). The remaining nodes arose via co-divergence with the species tree; these nodes are referred to as “speciation nodes”². In addition to the duplication, this reconciliation infers three losses, displayed as dashed lines. These three losses occurred on the branches leading to species *Glomerella*, *Histoplasma*, and *Inopilus*. The inferred association between ancestral genes and nodes in the species tree is indicated by the embedding of the gene tree in the species tree. Ancestral gene *u* is associated with the the root of the species tree, which is the ancestral species *Fungi*, labeled *F*. The divergence at ancestral gene *v* occurred on the branch from *Fungi* to *Dikarya*. By convention, if gene *g* is associated with a branch in the species tree, we say that *g* is associated with the species on the branch that this closest to the leaves. In this example, we say that *v* is associated with the ancestral species *Dikarya*, labeled *D*. In reality, the divergence at *v* may have occurred at any time between the divergence at *F* and the divergence at *D*.

Multiple optimal solutions: For the DL event model [22], the most parsimonious reconciliation is unique and easily calculated. In contrast, under the DTL model, there can be more than one minimum cost event history. For unit event costs, there are two most parsimonious reconciliations for the hypothetical family in Fig. 2.1. Both histories infer that the divergence at *w* is due to a horizontal gene transfer, indicated by an arrow, and that all other nodes are speciation nodes; both also infer one gene loss. The histories differ in the direction of the transfer, the location of the loss in the gene tree, and the species in which the loss occurred. In Fig. 2.1(d), the transfer occurs from *Histoplasma* to *Inopilus*. Node *v* is associated with *Ascomycota* and a loss occurred in *Inopilus*. In Fig. 2.1(e), the transfer

²In NOTUNG-2.8 and NOTUNG-DM, nodes that are not explicitly annotated as duplications or transfers are speciation nodes.

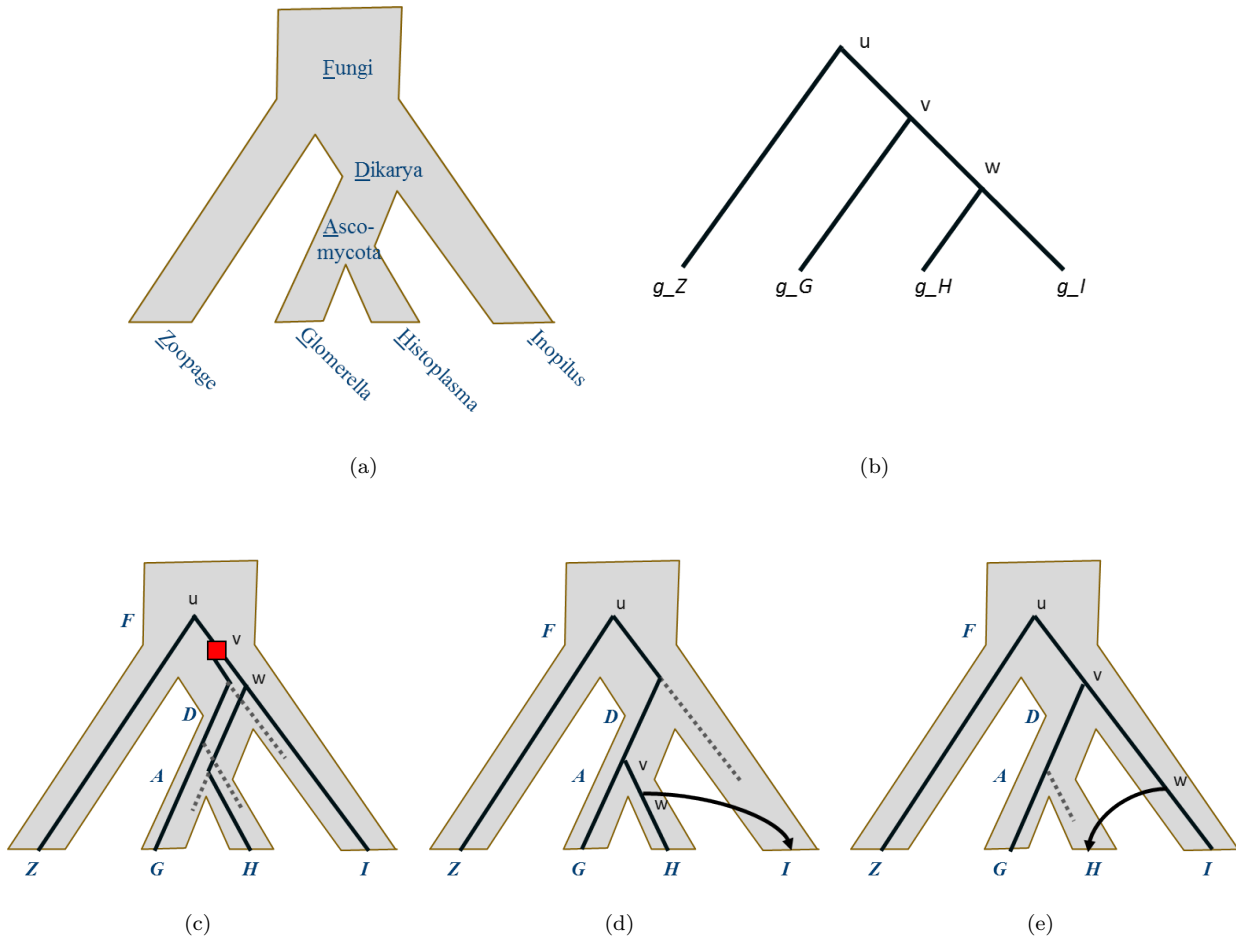


Figure 2.1: Gene tree species tree reconciliation. **(a)** A species tree, S , with four fungal species. **(b)** A gene tree, G , for a hypothetical gene family with one gene in each species in S . **(c)** Reconciliation of gene tree G with species tree S with the DL event model. This reconciliation requires one duplication (red square) and three losses (dashed lines) to explain the topological disagreement. **(d)** One of two equally parsimonious reconciliations of G with S under the DTL event model. This reconciliation requires one transfer (arrow) and one loss (dashed line) to explain the topological disagreement. **(e)** A second reconciliation of G and S with one transfer and one loss. This reconciliation differs from the history in (d) in the location of the events relative to both the gene and species trees.

occurs from *Inopilus* to *Histoplasma*. Node v is associated with *Dikarya* and a loss occurred in *Histoplasma*.

Both histories have a transfer between *Histoplasma* and *Inopilus*; they differ in the direction of the transfer, but not in the two species involved in the transfer. One way to interpret this result is to say that the reconciliation infers the species that participated in the transfer, but cannot determine in which direction the transfer occurred. This is not always true; there are cases where one direction is preferred over the other. For example, when gene tree $((g_Z, g_G), g_H), g_I$ is reconciled with the species tree in Fig. 2.1(a), there is a single optimal history with a transfer from *Glomerella* to *Zoopage*.

The number of optimal solutions depends on the gene event costs. For the example in Fig. 2.1, two events (one transfer and one loss) are required to explain the disagreement under the DTL model; four events (one duplication and three losses) are required under the DL model. Given unit costs, the history in Fig. 2.1(c), with four events, is sub-optimal under the DTL model.

Given the default costs in NOTUNG-2.8 ($C_D = 1.5$, $C_T = 3.0$ and $C_L = 1.0$), the reconciliations with one transfer and loss (Figs. 2.1(d) and 2.1(e)) both have an event cost of 4.0. The reconciliation with one duplication and three losses (Fig. 2.1(c)) has an event cost of 4.5 and is sub-optimal under the DTL model with these costs as well. If the transfer cost is increased to $C_T = 3.5$, then all three histories have an event score of 4.5. In this case, there are three optimal reconciliations. If the transfer cost is increased further ($C_T > 3.5$), then there is only one optimal reconciliation (Fig. 2.1(c)).

Temporal feasibility: Reconciliation with event models that include transfers is complicated by the fact that transfers introduce additional temporal constraints: a transfer can only occur between species that co-exist. These constraints are superimposed on the temporal constraints that are inherent in phylogenetic trees: a taxon cannot have predated its ancestors. These different types of constraints can combine to produce infeasible scenarios, in which there is no ordering of events that is consistent with a timeline. The challenge is to find event histories that are both parsimonious and temporally feasible.

Unfortunately, the problem of inferring optimal, feasible event histories has been shown to be NP-complete [7]; that is, to belong to a class of problems that are provably intractable with current mathematical knowledge. Consequently, the only guaranteed way to find the most parsimonious event history is to consider all possible histories (even those that are not most parsimonious), test each history for temporal feasibility, and then output the lowest cost, feasible solution(s). Since enumerating all possible histories is clearly impossible for all, but the smallest problem instances, the problem has been addressed by various authors either by imposing additional constraints that prevent the occurrence of infeasible histories, or by using heuristics that generally infer histories that are both optimal and feasible, but do not guarantee to find such a history for all possible inputs.

NOTUNG-2.8 [18] employs a heuristic approach in which all histories that minimize the event cost are generated and tested to determine whether they are temporally feasible. If a history generated in this manner is feasible, then it is also optimal. NOTUNG-2.8 reports all

optimal, temporally feasible event histories. However, if all of the candidate histories prove to be infeasible, then there is no known algorithm for finding minimum cost event histories that are also feasible, other than exhaustive search.

2.2 An event-based model of multidomain evolution

Evolutionary analysis of multidomain families is challenging because different parts of the sequence can have different histories. Because of their mosaic structure, multidomain families are not amenable to current methods of phylogenetic analysis, which assume that all residues in a sequence have the same history. In contrast, multidomain families often contain proteins with different domain compositions. Moreover, different regions of the same protein may have different evolutionary histories.

A practical barrier to applying current methods is that most require a multiple sequence alignment (MSA) as input. It is not, in general, possible to construct a full length alignment of a protein family with varied domain composition. A more fundamental problem is that current methods do not model of the events that modify domain architectures. Hence, these methods are not equipped to take domain events into account in inferring the tree that best explains the data.

These challenges are addressed in NOTUNG-DM by an domain event inference algorithm [19] for multidomain families evolving according to a *locus model* [17], in which novel domain arrangements arise through internal duplication, loss, and insertion of domains into an existing gene. The assumptions of the locus model justify the premise that the history of the family as a whole can be described by a tree. Many, if not most, families do, in fact, evolve according these assumptions. This assumption is consistent with the existence of promiscuous domains that lend themselves to insertion in new chromosomal environments [1, 11, 13, 14] and reports of young genes that arose through duplication of existing genes, followed by acquisition of additional domains [3, 8, 9, 16, 20]. Moreover, domain insertion into an existing gene is more likely to be viable since all regulatory and termination signals required for successful transcription are present.

Multidomain families evolve on three levels of organization: domains, genes, and species. Our model of this three-level process is illustrated in Fig. 2.2, which depicts the evolutionary history of a hypothetical multidomain family that consists of four genes, $k_1 - k_4$, in two Ascomycete genomes. The members of the k gene family contain various combinations of three domain families, blue (rounded rectangles), red (circles), and green (sharp rectangles). All four, present-day k genes possess a blue domain, as well as some combination of green and red domains. Fig. 2.2 presents two different views of the evolution of this family. In Fig. 2.2(a), domains and genes are represented as segments of genomes. Gene and domain events displace, delete or generate new copies of those segments. In Fig. 2.2(b), the same evolutionary history is depicted as domain trees embedded within a gene tree, which is in turn embedded in a species tree.

Events on the domain, gene, and species levels all contribute to the organization of the present-day members of this family. The common ancestor of the family is a single gene in

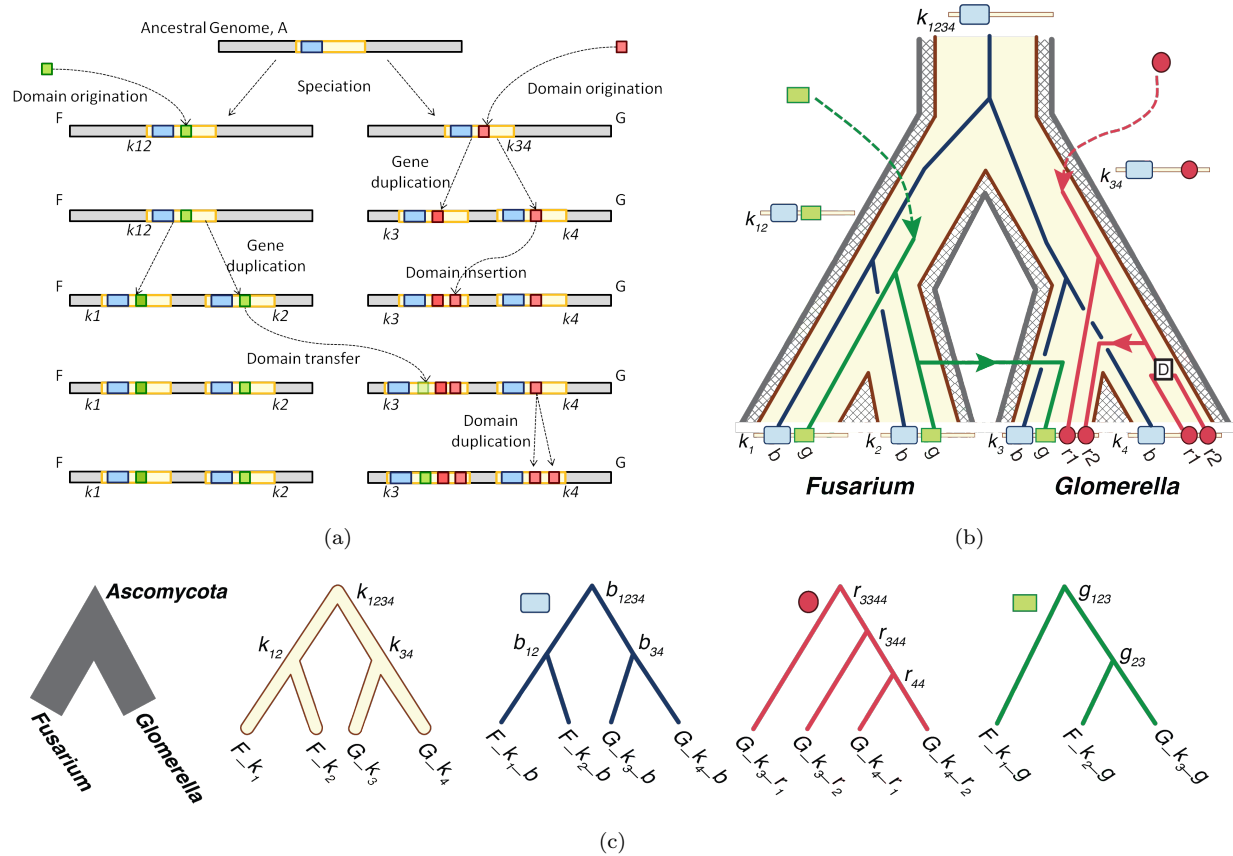


Figure 2.2: This figure shows co-evolution on three different levels of organization: species evolution, gene family evolution and domain shuffling. **(a)** Evolution of a hypothetical multidomain gene family in two genomes, F and G . Grey bars represent the chromosomes, cream rectangles represent genes, and colored boxes represent domains. **(b)** The evolutionary progression of the genes encoding the family, with embedded domain trees, is shown in the context of the phylogeny of the species that contains them. Domain trees are represented by thin trees colored according to the domain they represent. The gene tree is the “fat” cream colored tree. The species tree is the “fat” hatched filled tree. **(c)** Species, gene, and domain trees.

the Ascomycete ancestor (Genome A). This gene possesses a single domain, a member of the blue domain family. On the species level, a single event occurred: a speciation resulting in two lineages leading to the present-day *Fusarium* (F) and *Glomerella* (G) lineages. The resulting species tree has two leaves. On the gene level, the four present-day loci arose via co-divergence with the speciation event that gave rise to genomes F and G , followed by two gene duplications, one in the lineage leading to F and the other in the lineage leading to G .

On the domain level, the blue domain was present in the common ancestor of the family and evolved with the locus by vertical descent. Therefore, the topology of the blue domain tree agrees with the topology of the gene tree. The green domain in gene k_{12} and the red domain in gene k_{34} arose through independent domain acquisitions, after the separation of the F and G lineages. Three subsequent events gave rise to the present-day domain architectures: a domain transfer, a domain insertion, and a domain duplication. Unlike the blue domain tree, the red and green domain family trees have topologies that differ from that of the gene tree.

This example illustrates several key features of the locus model. First, new domain architectures arise through insertion of an auxiliary domain *into an existing gene*. This assumption allows us to decouple the concepts of a homologous sequence and a homologous gene (i.e., a locus or segment of a genome that encodes a protein.) Homologs are “characters that have descended, usually with divergence, from a common ancestral character.” [5]. In the locus model, genes that are descended from the same ancestral locus are homologous. Treating the locus as a character allows us to extend the definition of a homologous gene family to multidomain families, even though they may contain different domains with different histories. Further, the evolutionary history of a family defined in this way is the history of the gene locus, which can be represented by a rooted, binary tree.

NOTUNG-DM extends the reconciliation framework to three levels of organization. This requires that the evolutionary history of domains, genes and species can all be represented by rooted binary trees. In our framework, the evolutionary history of entities at each level of organization (domains, genes, or species), when considered in isolation, can be modeled by rooted, binary trees (Fig. 2.2(c)). The history of the family as whole is represented by the superposition of domain, gene and species trees, where some edges in the “symbiont” tree may be horizontal when compared with the “host” tree. In other words, the family history is modeled as a binary tree with cross edges. Note that this is distinct from a reticulate network, in which a node can have two parents.

A further assumption of our model is that domain insertions and transfers only involve domains within the same gene family. In other words, for a given domain family, we assume that the domain instances that appear in the gene family under consideration form a clade in the domain tree.

2.2.1 Inferring domain shuffling with reconciliation

Multidomain evolution is driven by molecular processes, including non-allelic homologous recombination, retrotransposition, read-through errors, and unequal crossing over. For the purposes of evolutionary reconstruction we model these processes with a set of abstract

events: domain duplication, domain loss, and horizontal acquisition of a domain by transfer or insertion. More formally, we define the *Duplication-Transfer-Insertion-Loss (DTInL)* domain event model as follows:

Co-divergence (\mathcal{C}) A bifurcation in the domain tree that arose through a bifurcation in the gene tree. The gene tree bifurcation may have arisen via speciation (\mathcal{C}_S), gene duplication (\mathcal{C}_D), or gene transfer (\mathcal{C}_T).

Duplication (\mathcal{D}) A single domain is copied, resulting in two separate copies of the domain within the same gene.

Loss (\mathcal{L}) A domain is deleted from the gene (and genome).

Domain insertion (\mathcal{I}) A new copy of the domain is inserted into a different gene within the *same* genome.

Horizontal domain transfer (\mathcal{T}) A new copy of a domain is inserted into a gene in a *different* genome.

We also consider a restricted *Duplication-Insertion-Loss (DInL)* domain event model that allows domain insertion, but not horizontal transfer of genes or domains. This restricted event model is appropriate for analyzing multidomain family evolution in species in which horizontal gene transfer rarely or never occurs.

NOTUNG-DM takes as input a rooted gene tree that has been reconciled with a species tree and a rooted domain tree, where every leaf in the domain tree is associated with a leaf in the gene tree. These trees must be provided by the user.

Given a set of amino acid sequences, domains can be identified using a domain database such as PFam, Superfamily, or CDD [4, 6, 10]. Once the subsequences corresponding to the domain family have been extracted, standard molecular phylogenetic methods can be used to reconstruct a domain tree.

The history of a gene family, as defined in our model, is the history of the gene locus, which can be modeled as a binary, rooted tree. Although this is conceptually straightforward, currently, there are no widely-accepted approaches to determining the history of a locus that has been undergoing domain shuffling (i.e., where different parts of the sequence had different histories.)

One approach to estimating a gene tree under these circumstances is to use the history of one of the constituent domains as a “proxy” for the history of the gene family. the locus. Many multidomain families arise from a single-domain progenitor gene [2, 12, 15], which undergoes duplication, followed by domain insertion, resulting in the progenitors of subfamilies with different domain architectures (e.g., Fig. 2.2(a)). These subfamilies then further expand through additional duplications. For families that follow this pattern, the evolutionary history of the single domain (called the “primary” domain) in the progenitor sequence is congruent with the history of the locus and can be used as a proxy gene tree.

However, in general, there is no guarantee that a domain that is present in every member of the gene family was present in the ancestor and evolved by vertical descent. We

propose a set of criteria for evaluating whether a domain can be used as a proxy: (1) evidence of vertical descent based on conserved synteny; (2) appearance of the domain in only one family; and typically in the same number of domain copies within the family; (3) absence of traits characteristic of mobile domains, such as short amino acid sequences or 1-1 intron phase. When multiple domains in the same family fit these criteria, we also require that the set of “primary” domains have congruent trees.

A second approach to estimating the history of a multidomain locus is to estimate the history of neighboring genes. This approach can be used when the gene content of the regions flanking the genes of interest are conserved.

The output is a reconciled domain tree with an event history that minimizes the domain event score

$$\kappa = \kappa_{\mathcal{D}}n_{\mathcal{D}} + \kappa_{\mathcal{T}}n_{\mathcal{T}} + \kappa_{\mathcal{I}}n_{\mathcal{I}} + \kappa_{\mathcal{L}}n_{\mathcal{L}},$$

where n_{ϵ} is number of occurrences of domain event ϵ in the reconciliation and κ_{ϵ} is the cost of ϵ . Event costs are specified by the user and reflect the relative importance between different event types. Note that the cost of co-divergences is always zero. The reconciliation procedure also yields the association between ancestral domains, ancestral genes, and ancestral species. This information can be used to reconstruct the timing of those events relative to gene and species divergences, as well as ancestral domain content.

Reconciliation algorithms for reconstructing domain shuffling entail challenges similar to those encountered in gene tree - species tree reconciliation with a DTL model, including multiple solutions and temporal constraints. Like NOTUNG-2.8, NOTUNG-DM reports all optimal, temporally feasible domain shuffling event histories.

2.2.2 Comparison with Wagner parsimony

In the past, parsimony approaches have been applied to multidomain families to gain a superficial understanding of the impact of domain shuffling. In the typical domain architecture (DA) model, a multidomain sequence is treated as a set or sequence of “tokens” (e.g. domain names or ids) representing its domain composition. For example, gene k_3 in Fig. 2.2 would be represented as blue-green-red-red. In this model, domain content is treated as multinomial character data; m instances of a domain are present in a given architecture, where m is a non-negative integer. Gene k_3 has 1 blue, 1 green, and 2 red domains. Given a tree with architectures on the leaves, the ancestral state at node v is inferred by minimizing the number of domain gains and losses between v and its children using Wagner or Dollo parsimony. Here, the ancestral state is the number of instances of each domain. The advantage of DA parsimony is that it is easy to implement and runs quickly. A disadvantage is that it does not contain an explicit model of domain evolution; it cannot distinguish between domain duplication, insertion, transfer, and loss. In addition, it is susceptible to the inherent problem of all parsimony approaches on character data: a failure to recognize parallel gains and/or losses.

As demonstrated (Fig. 2.3) for our hypothetical gene family, the DA approach may underestimate the number of gains and losses in a gene family. First, it infers only three

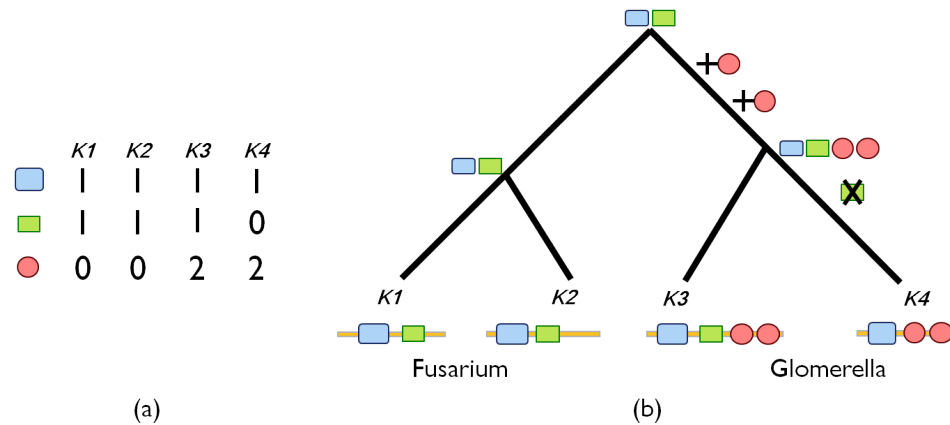


Figure 2.3: (a) A character state table showing the number of copies of the blue (rounded rectangle), red (circle), and green (sharp rectangle) domains in each gene. (b) The history of domain gains and losses as inferred using Wagner parsimony. The green domain is present in the root and is lost once during the evolution of the gene family. The red domain is gained twice before the divergence of k_3 and k_4 . Note that Wagner parsimony underestimates the number of events, compared with the history based on reconciliation.

events: two gains of a red domain and one loss of a green domain. In contrast, in the true history, there were five events: the original gain of a red domain, followed by one insertion and one duplication, and one original gain of a green domain, followed by one transfer. Second, the ancestral states are different from those of the true history. The DA parsimony model predicts that the least common ancestor of the gene family had one blue and one green domain, but no red domain. The true history, however, has only one blue domain, and no green or red domains.

In contrast, our reconciliation-based approach captures sequence variation across domain instances and is based on an explicit event model.

Chapter 3

Worked Examples

In this chapter, NOTUNG-DM's primary functions are illustrated with two worked examples. These examples introduce NOTUNG-DM's I/O formats and demonstrate how to reconcile domain trees. NOTUNG-DM requires a species tree, a gene tree, and a domain tree as input. These trees must be provided by the user. The trees used in the examples in this chapter are provided with the NOTUNG-DM distribution, in the `sampleTrees` folder. Given the association between leaf nodes, NOTUNG-DM establishes (1) the association between each ancestral domain (i.e., each internal node in the domain tree) and nodes in the gene and species trees and (2) the most parsimonious history of events that explain this association.

The resulting reconciled trees are presented graphically in the NOTUNG-DM GUI for exploratory analysis. This GUI facilitates tree visualization, enabling the user to inspect duplicated, inserted, transferred, and lost nodes in a tree, and color annotate genes for visual differentiation or presentation. Reconciled domain trees can be saved in Notung format and read into NOTUNG-DM later (see [Sec. 3.3.1](#)), or they can be saved as images in .PNG format (see Notung-2.8 Manual [Sec. 3.3](#) for more information). In addition, NOTUNG-DM provides a textual summary of the history of inferred events, which can be saved in .txt format (see [Sec. 3.3.4](#)).

Information presented in the GUI can be used to calculate the number of domain copies in each ancestral gene and species, as well as the branches in the gene and species trees where events occurred. The history of each domain family is inferred separately. The current version of NOTUNG-DM does not combine individual domain histories to reconstruct ancestral domain architectures. However, it does generate the information required to enable the user to reconstruct the ancestral architectures in a post-processing step. We discuss how to interpret the results to create ancestral reconstructions in [Chapter 4](#).

3.1 Example 1

The first example is gene family k , the hypothetical multidomain family introduced in [Fig. 2.2](#), which contains three different domain families. The history of each domain can be considered independently. Input trees for this worked example are provided in the `Example_1`

folder.

Getting Started After you have downloaded the NOTUNG-DM distribution, launch the NOTUNG-DM executable, as described in [Section 1.3](#). To introduce the NOTUNG-DM GUI, we start by opening some trees.

Open the tree files

1. Click “**File → Open Species Tree**”
2. In the Open dialog box, select the `sampleTrees` folder, followed by the `Example_1` folder. Select the tree file `speciestree1.nwk` and click “**Open**”.

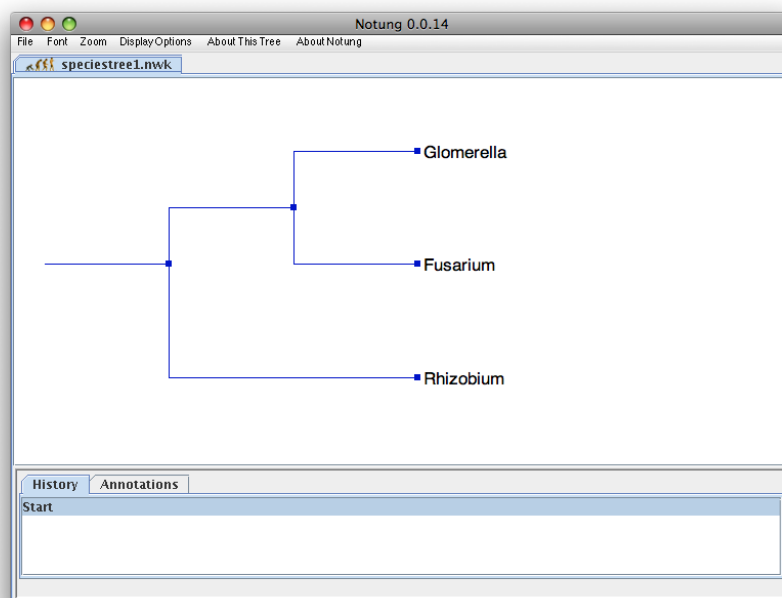


Figure 3.1: The opened species tree displayed in NOTUNG-DM.

Once loaded, the species tree appears in the tree panel ([Fig. 3.1](#)). Leaf node names are displayed by default.

Labels may be provided for internal nodes; this is not required, but is useful for reference — if internal labels are not given, the nodes are assigned alphanumeric labels (such as *n1*, *n2*, etc.). See [Fig. 3.4](#) for an example of how to include internal node labels in Newick format. To display the internal node labels, use the “**Display Options**” pull down menu at top of the NOTUNG-DM window as shown in [Fig. 3.2](#).

3. Click “**Display Options → Display Internal Node Names.**”

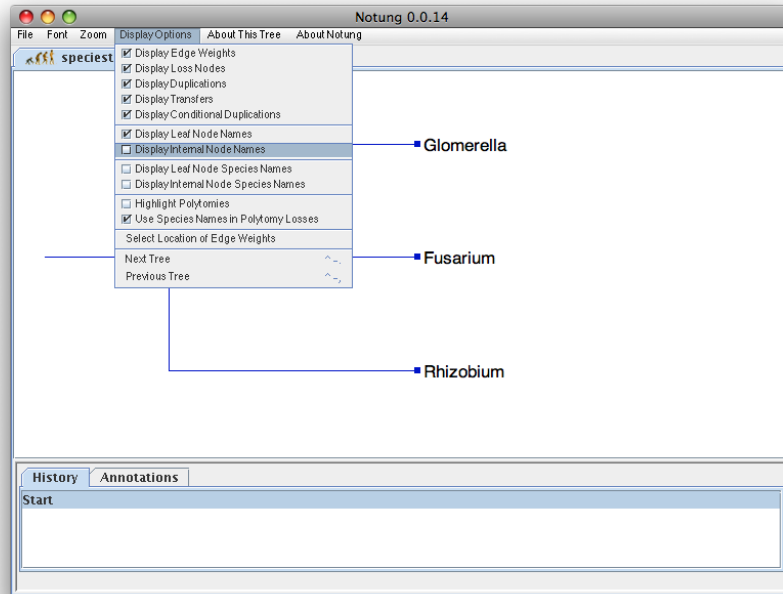


Figure 3.2: The menu available for displaying internal node names in NOTUNG-DM.

The names/identifiers of all internal nodes, i.e., ancestral species, should now appear on the species tree (Fig. 3.3(a)).

4. Click “**File** → **Open Gene Tree**” and open `genetree_K.nwk`.
Once loaded, the gene tree is displayed in the tree panel.
5. Click “**Display Options** → **Display Internal Node Names**” to view the internal node labels.

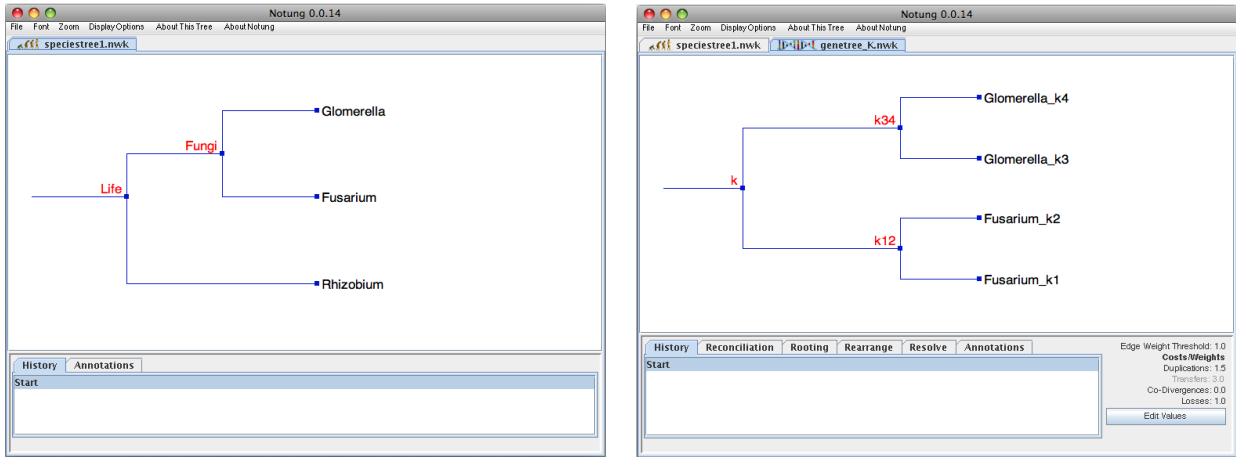
The gene tree is now displayed with internal node names (Fig. 3.3(b))¹.

6. Click “**File** → **Open Domain Tree**” and open `blue.nwk`
Once loaded, the blue domain tree is displayed in the tree panel.
7. Click “**Display Options** → **Display Internal Node Names**” to view the internal node labels.

The blue domain tree is now displayed with internal node names (Fig. 3.3(c)).

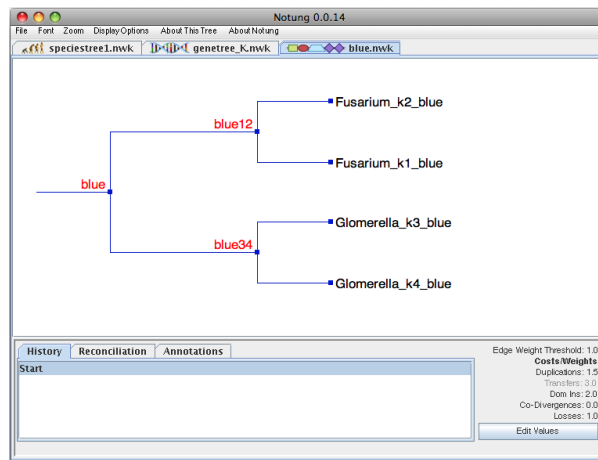
When first launched, the NOTUNG-DM program window will be blank. Fig. 3.3 shows the NOTUNG-DM graphical interface once a domain, gene, and species tree have been opened.

¹Note that internal node names have been provided for all gene and domain trees here. The names were generated based on the labels of the nodes children/descendants. For example, gene node `k12` is so named because it is the parent of nodes `Fusarium_k1` and `Fusarium_k2`



(a) Species tree

(b) Gene tree



(c) Blue Domain Tree

Figure 3.3: Species, gene, and blue domain trees prior to reconciliation

Within the GUI, each tree is presented in a separate view, identified by a tab (as seen in Fig. 3.3) that displays the associated filename and an icon that indicates whether the tree is a domain tree (a multidomain architecture), gene tree (a double helix), or species tree (the evolution of humankind). The user can move from one tree to another by clicking on a tab to select the corresponding tree.

In NOTUNG-DM, the command that was used to open the tree determines whether NOTUNG-DM treats a tree as a domain, gene, or species tree. NOTUNG-DM cannot determine the type of tree automatically from information in the tree file. If a tree is opened with the wrong command, e.g., if a domain tree is opened using the “Open Gene Tree” function, NOTUNG-DM will interpret that tree as a gene tree and incorrect behavior will result. If NOTUNG-DM does not do what you expect, check to make sure that each tree is labeled with the appropriate icon. If not, simply close the tree (“Ctrl+W”) and reopen it with the

correct command².

The NOTUNG-DM GUI has several components. The tree that is currently selected appears in the *tree panel*, the white rectangular area in the upper part of the window. Pull-down menus above the tree panel allow the user to modify the appearance of the tree and to obtain additional information about its properties (see Notung-2.8 Manual Chap. 11 for more information). NOTUNG-DM functions are invoked from the *task panel*, the grey, rectangular area below the tree panel. The specific functions that NOTUNG-DM can perform on each kind of tree are given in [Table 1.1](#).

The tabs at the top of the task panel correspond to the various tasks that NOTUNG-DM can perform. Clicking on a tab puts NOTUNG-DM in the corresponding task mode, revealing the buttons that perform functions that are specific to that mode. The set of functions that NOTUNG-DM can perform depends on the type of tree that is currently selected. Two task modes, **History** and **Annotations**, are applicable to all tree types (see Notung-2.8 Manual Chap. 9 and 10 for more information). Domain and gene trees also have a **Reconciliation** task mode. Additional task modes are available for gene trees. In general, for gene trees, NOTUNG-DM can perform most of the same functions as NOTUNG-2.8. These are described in detail in the Notung-2.8 Manual and will not be described here, except when those functions are required for functions specific to multidomain evolution.

NOTUNG-DM uses an Event Score to score inferred event histories. The **Event Score** is defined to be:

$$\kappa = \kappa_{\mathcal{D}}n_{\mathcal{D}} + \kappa_{\mathcal{T}}n_{\mathcal{T}} + \kappa_{\mathcal{I}}n_{\mathcal{I}} + \kappa_{\mathcal{L}}n_{\mathcal{L}},$$

where n_{ϵ} is number of occurrences of event ϵ in the reconciliation and κ_{ϵ} is the cost of ϵ . When a tree is reconciled, NOTUNG-DM displays the Event Score of the reconciled tree, as well as the number of, Duplications (\mathcal{D}), Transfers (\mathcal{T}), Insertions (\mathcal{I}), and Losses (\mathcal{L}) in the **Status Bar** at the bottom-left corner of the program window. Event costs are specified by the user and reflect the relative importance of different event types. The default values are 1.5 for duplications, 3.0 for transfers, 2.0 for insertions, and 1.0 for losses. Note that the cost of co-divergences is always zero and that the cost of insertions is only relevant when reconciling domain trees. Parameter values are displayed in the bottom-right corner of the program window under the title **Costs/Weights**. These values can be changed by the user by clicking the “**Edit Values**” button. The values for the insertion and transfer costs can only be edited if the “Infer Insertions” or “Infer Transfers” option, respectively, is selected. Note that when a species tree is selected, the program window does not display the parameter values. For these examples, we will use the default event costs.

Input Formats. In order to infer domain shuffling events, NOTUNG-DM requires a domain tree, a gene tree, and a species tree. These trees, which must be rooted and binary, are provided by the user. They may be in Newick, NHX, or Notung format. The trees used in Example 1 are shown in Newick format in [Fig. 3.4](#).

The species tree must contain all the species represented in the gene tree, which, in turn must contain all the genes represented in the domain tree. The gene tree may contain

²To close a tree: Select the tree to close; then select “**Close**” from the “**File**” menu.

```

Species tree:
    ((Fusarium,Glomerella)Fungi,Rhizobium)Life;

Gene tree:
    (
      (Glomerella_k3,Glomerella_k4)k34,
      (Fusarium_k1,Fusarium_k2)k12
    )k;

Blue domain tree:
    (
      (Fusarium_k1_blue, Fusarium_k2_blue)k12.blue,
      (Glomerella_k4_blue,Glomerella_k3_blue)k34.blue
    )k.blue;

```

Figure 3.4: Species, gene, and blue domain trees in Newick format.

additional genes not represented in the domain tree; the species tree may contain additional species as well. To ignore these additional taxa, the pruning option should be used (see Notung-2.8 Manual Sec. 5.1 for more information).

The association between present-day domains, genes and species must be encoded in the input trees. NOTUNG-DM uses this information to infer the association between ancestral nodes in the gene, species, and domain trees. Leaf nodes must be labeled with the information required to determine (1) the species from which each leaf taxon in the gene tree was derived and (2) the specific gene from which each leaf taxon in the domain tree was derived. This is achieved by embedding the species name in the gene leaf label and the gene and species names in the domain leaf label in the prescribed format described below. This labeling scheme encodes the association between the present-day domain, gene and species.

For species trees, each leaf must be labeled with a unique alpha-numeric string that acts as a species identifier. Spaces and underscores are not allowed. In gene trees, each leaf label must contain a gene identifier, preceded by the identifier of the species in which the gene is found; the combination of gene and species must be unique. Leaves in domain trees must be labeled with a domain identifier, preceded by the identifiers of both the gene and species in which the domain is found. The required formats are summarized in Table 3.1. For example, the label for the *k1* gene in the gene tree in Fig. 3.3(b) is *Fusarium_k1*, where *Fusarium* is the species containing gene *k1*. Similarly, the domain tree label for the blue domain in gene *Fusarium_k1* is *Fusarium_k1_blue*.

Species tree	Species
Gene tree	Species_GeneID
Domain tree	Species_GeneID_DomainID

Table 3.1: Leaf label name formats for reconciling domain trees.

Reconciling the gene tree with the species tree: NOTUNG-DM infers domain event histories by reconciling the domain tree with a gene tree that has been reconciled with the associated species tree, so we must reconcile the gene tree before proceeding. If the user attempts to reconcile a domain tree with an unreconciled gene tree, an error message will appear.

Reconcile the gene tree

1. Click on the `genetree.K.nwk` tab to select the gene tree.
2. Click the “**Reconciliation**” tab in the task panel.

The **Reconciliation** task panel appears below. By default, NOTUNG-DM reconciles gene trees with species trees using a DL event model. A check box in the Reconciliation task panel allows users to add transfers to the event model.

3. Check the “**Infer Transfers**” box.

In the parameter values area to the right, the text “Transfers: 3.0” switches from gray to black, indicating that the DTL model will be used for reconciliation, with the default transfer cost of 3.

4. Click “**Reconcile/Rereconcile.**”

The Reconciliation dialog box appears. In this dialog box, NOTUNG-DM asks you to specify (1) the species tree to use for the reconciliation and (2) the naming convention used in the gene tree to indicate the species associated with each gene.

5. Select `speciestree1.nwk` in the drop-down menu labeled “Please select a species tree to reconcile with.”

Currently, the only selection available is `speciestree1.nwk`. However, if you have more than one species tree open in NOTUNG-DM, you must specify here which species tree to use.

6. Under the section labeled “Specify Species Label,” select “**Prefix of the gene label.**”

This section in the dialog box asks you to specify the naming convention used in the gene tree to indicate from which species the genes originated. For gene tree - species reconciliation, NOTUNG-DM allows several different naming conventions, which are

described in Notung-2.8 Manual Appendix A. However, only the prefix convention is allowed when the gene tree is being reconciled for later use with domain tree reconciliation.

7. In the dialog box, click “**Reconcile.**”

The reconciled gene tree now appears in the tree panel (Fig. 3.5). Gene events are displayed on the reconciled tree. The Event Score of the reconciled tree is displayed in the status bar in the bottom-left corner of the program window. With default costs, the Event Score is 3.0.

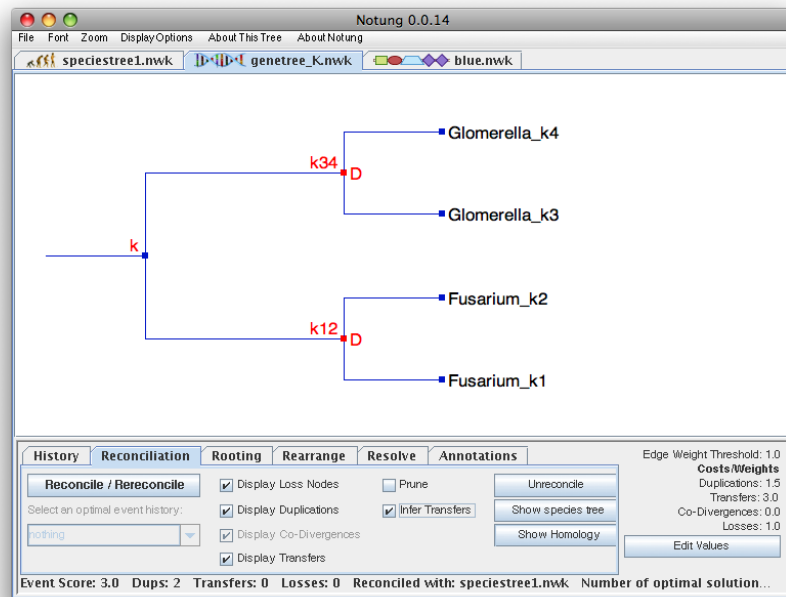


Figure 3.5: The reconciled gene tree.

Once the gene tree has been reconciled, a species tree node is associated with each gene tree node. To understand when events occurred in the relation to the species tree, it is useful to display this association/mapping.

8. Click “**Display Options** → **Display Internal Node Species Names**” to view the names of the species associated with the internal gene tree nodes.

The reconciled gene tree now shows the species tree nodes associated with each internal gene tree node (Fig. 3.6).

Inspection of the events on the reconciled gene tree show that the root of the gene tree is a speciation node, i.e., a co-divergence with a bifurcation at Fungi in the species tree, and is represented by a small blue square. Two red D’s in the tree indicate two inferred duplications: the $k12$ divergence arose via a duplication in *Fusarium* and the $k34$ divergence

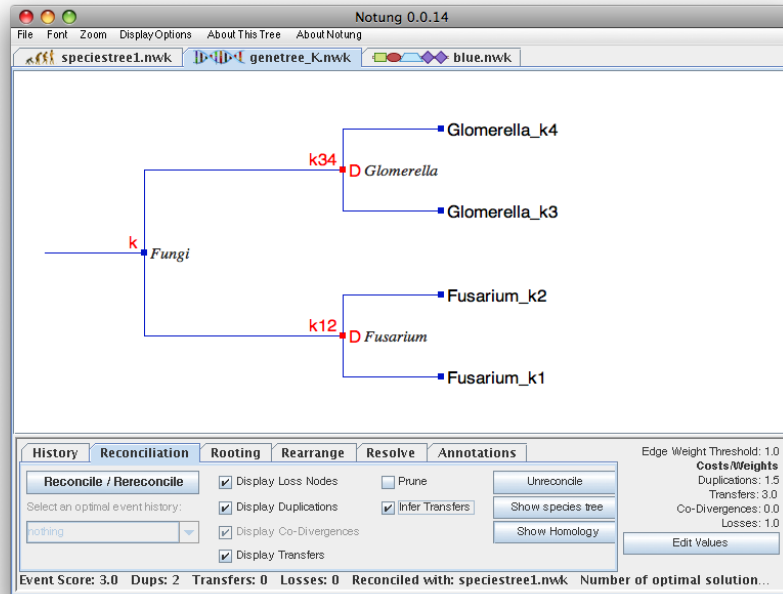


Figure 3.6: The reconciled gene tree displaying the species associated with each internal gene node.

arose via a duplication in *Glomerella*. Note that this consistent with the “true” history shown in Fig. 2.2.

Image	Event	Text	Color	Description
	Gene Duplication	‘D’	red	square
	Speciation	—	blue	square
	Gene Transfer	‘T’	yellow	triangle; edge highlighted
	Gene Loss	“*G*LOST”	gray	leaf node

Table 3.2: Event display in NOTUNG-DM for a reconciled *gene* tree. Inferred duplication and speciation events are indicated on the internal node representing the bifurcation caused by such an event. Inferred horizontal gene transfer events are represented on tree branches, indicating the bifurcation that caused the event and the recipient of the event. The triangle on this event is clickable. When clicked, the donor and recipient species of that event are displayed. Inferred gene loss events are displayed as gray external (leaf) nodes, indicating the species in which the loss occurred.

Reconciling the blue domain tree with the gene tree: Now that the gene tree has been reconciled with the species tree, the domain trees can be reconciled with this reconciled

gene tree to infer domain events.

Reconcile the blue domain tree

1. Click on the `blue.nwk` tab to select the blue domain tree.
2. Click the “**Reconciliation**” tab.

The domain tree **Reconciliation** task panel appears. By default, NOTUNG-DM reconciles domain trees using an event model with duplications, insertions, and losses (DInL). Note that the list of weights in the Parameter Values area now includes a cost for domain insertions (“**Dom Ins**”), which was not present in the gene tree task panel (compare with Fig. 3.3(b)). The default cost is 2.0.

A check box in the Reconciliation task panel allows the user to add domain transfers to the event model.

3. Check the “**Infer Transfers**” box.

The text “Transfers: 3.0” in the “Costs/Weights” area to the right switches from gray to black, indicating that the DTInL event model, with the default transfer cost of 3.0, will be used.

4. Click the “**Reconcile/Rereconcile**” button.

A dialog box appears, as seen in Fig. 3.7.

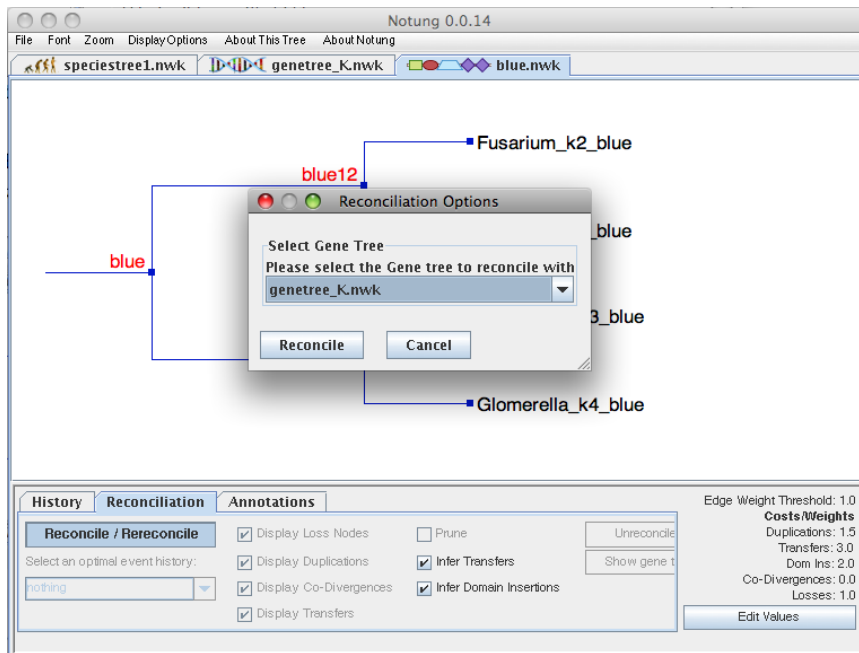


Figure 3.7: Dialog box for reconciling a domain tree with a gene tree.

In this dialog box, NOTUNG-DM asks you to specify which gene tree to use for the reconciliation. Note that this reconciliation dialog box is simpler for domain trees than for gene trees, because only one naming convention is allowed for reconciling domain trees. The only action required by the user is to select a gene tree.

- In the dialog box, select the correct gene tree in the drop-down menu.

Currently, the only selection available is `genetree_K.nwk`. However, if you have more than one gene tree open in NOTUNG-DM, you must specify here which gene tree to use. Note that the gene tree you select must have already been reconciled with a species tree. If you select an unreconciled gene tree, NOTUNG-DM will present an error message.

- In the dialog box, click “**Reconcile.**”

The reconciled domain tree now appears in the tree panel [Fig. 3.8](#). Domain events are displayed on the reconciled tree.

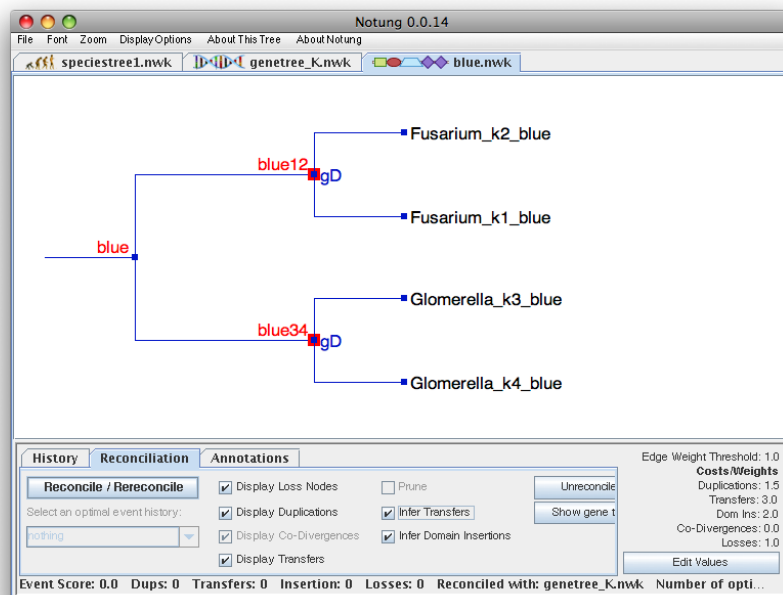


Figure 3.8: The reconciled tree for the blue domain.

Once the domain tree has been reconciled, one gene tree node and one species tree node are associated with each domain tree node. To understand when events occurred in the relation to the gene and species trees, it is useful to display these associations/mappings.

- Click “**Display Options** → **Display Internal Node Species Names**” to view the name of the genes and species associated with the internal domain tree nodes.

In a reconciled domain tree, in addition to the species, this function displays the gene associated with each internal domain tree node (Fig. 3.9). The ancestral associations are displayed together, separated by an underscore, as Species_GeneID.

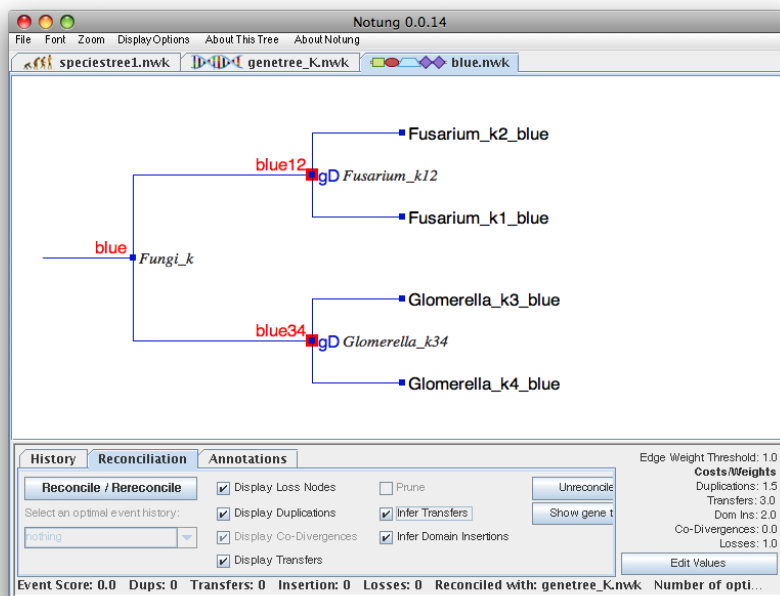


Figure 3.9: The reconciled blue domain tree displaying the gene and species associated with each internal domain node.

Following reconciliation, the domain tree is displayed with a mark up showing inferred domain events that impact the history of the domain. All nodes in the blue domain tree are co-divergences. This is because the blue domain has the same history as the gene locus (Fig. 3.3). As a result, the Event Score of this reconciled tree is 0.0, since co-divergences have zero cost.

In addition to domain events, the domain tree markup shows how gene events have shaped the topology of the domain tree. In the blue domain tree, the mark up indicates the gene event associated with each co-divergence. The root is a co-divergence with a speciation node in the gene tree and is represented by a small blue square. The other two internal nodes are co-divergences with gene duplications and are marked by a blue ‘gD’ and a small blue square nested in a larger red square. In each case, this indicates that the corresponding node in the gene tree is a duplication. In other words, an additional copy of the blue domain in a different gene arose through a gene duplication; this is distinct from a domain duplication, which would have resulted in an additional copy of the blue domain in the same gene.

Reconciling the red and green domain trees with the gene tree: Unlike the blue domain, which co-evolved with the gene locus by vertical descent, the red and green domain trees are topologically incongruent with the gene tree due to domain insertions, transfers,








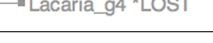
Image	Event	Text	Color	Description
	Domain Duplication	‘D’	red	square
	Gene Duplication	‘gD’	blue	square nested in large red square
	Gene Transfer	‘gT’	blue	triangle
	Speciation	—	blue	square
	Domain Transfer	‘T’	yellow	triangle; edge highlighted
	Domain Insertion	‘I’	purple	triangle; edge highlighted
	Gene Loss	“*G*LOST”	gray	leaf node
	Domain Loss	“*LOST”	gray	leaf node

Table 3.3: Event display in NOTUNG-DM. Inferred duplication and co-divergence events are indicated on the internal node representing the bifurcation caused by the event. Inferred transfers and insertions are represented on tree branches, indicating the bifurcation that caused the event and the recipient of the event. The triangle on such events is clickable; when clicked, the donor and recipient taxa of that event are displayed as gene and species pairs. Inferred loss events are displayed as gray external (leaf) nodes. The leaf label indicates the type of loss and the taxa in which the loss occurred.

and duplications. We demonstrate inference of those events by reconciling the red and green domain trees with the gene tree.

Reconcile the red domain tree

1. Click **“File → Open Domain Tree”**
2. In the Open dialog box, select the tree file `red.nwk` and open.
Once loaded, the red domain tree is displayed in the tree panel.
3. Click **“Display Options → Display Internal Node Names”** to view the internal node labels.
4. Click the **“Reconciliation”** tab and check the **“Infer Transfers”** box.
5. Click the **“Reconcile/Rereconcile”** button. In the dialog box, select `genetree_K.nwk` and click **“Reconcile.”**
The reconciled domain tree now appears in the tree panel. The Event Score of this reconciled tree is 3.5.
6. Click **“Display Options → Display Internal Node Species Names”** to view the genes and species associated with internal domain nodes.

The reconciled domain tree now shows the gene and species associated with each internal domain tree node (Fig. 3.10). The inferred history includes a domain duplication and a domain insertion. The root of the tree is associated with ancestral gene *k34* in the *Glomerella* genome. This indicates that the red domain was not present in the least common ancestor of the gene family. The event at this ancestral domain node is a co-divergence with a gene duplication. This divergence was followed by a domain insertion into gene *k3* of *Glomerella*. The insertion is represented by an edge highlighted in purple; a purple triangle, with a purple ‘I,’ appears halfway down this edge. The duplication, represented by a red node labeled with a red ‘D,’ occurred after the insertion, and resulted in two copies of the red domain in gene *k4* of *Glomerella*.

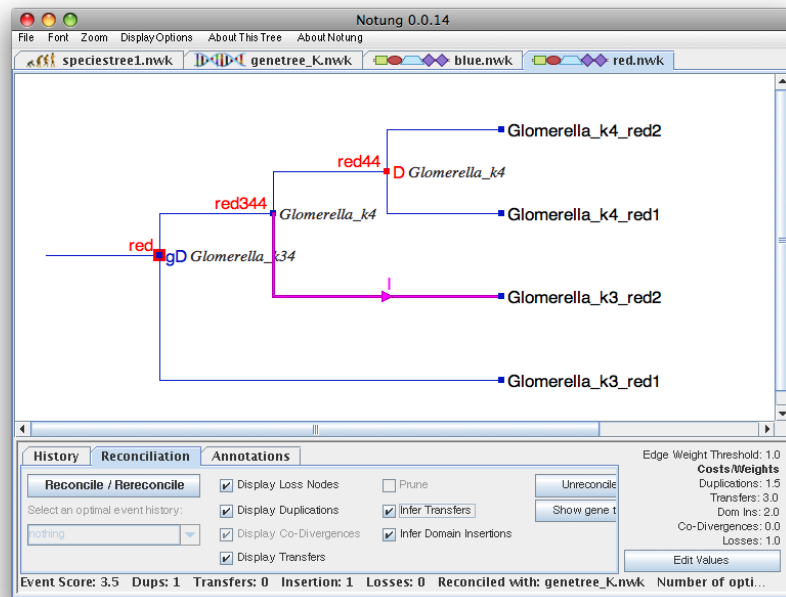


Figure 3.10: The reconciled tree for the red domain, displaying the gene and species associated with each internal domain node.

7. Click on the purple arrow.

The names of the donor and recipient genes appear (Fig. 3.11), showing that the inserted copy of the red domain in *Glomerella* gene *k3* originated from the gene *k4* in *Glomerella*.

Reconcile the green domain tree

1. Click “File → Open Domain Tree”
2. In the Open dialog box, select the tree file `green.nwk` and open.
Once loaded, the green domain tree is displayed in the tree panel.

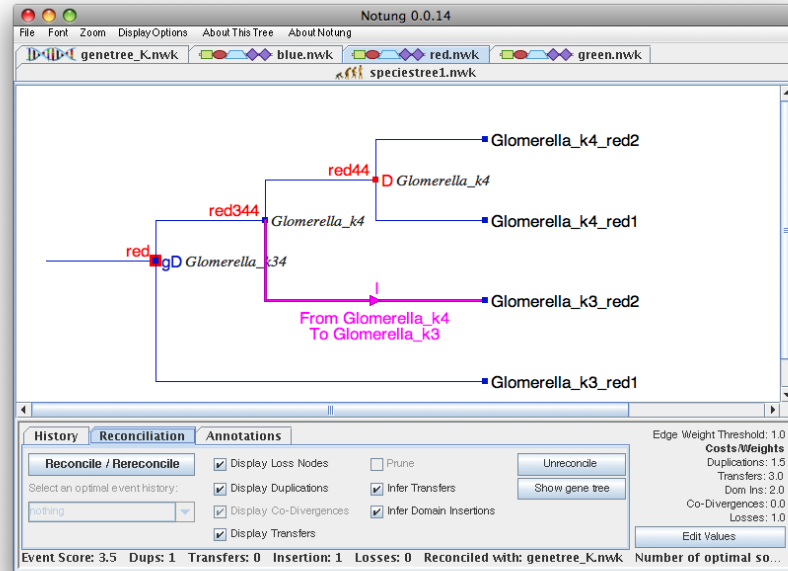


Figure 3.11: The insertion event in the red domain, displaying the donor and recipient genes and species.

3. Click “**Display Options** → **Display Internal Node Names**” to view the internal node labels,
4. Click the “**Reconciliation**” tab and check the “**Infer Transfers**” box.
5. Click the “**Reconcile/Rereconcile**” button. In the dialog box, select `genetree_K.nwk` and click “**Reconcile.**”

The reconciled domain tree now appears in the tree panel. The Event Score of this reconciled tree is 3.0, the cost of a single transfer.

6. Click “**Display Options** → **Display Internal Node Species Names**” to view the gene and species associated with each internal domain node.

The reconciled domain tree now shows the associated gene and species tree nodes (([Fig. 3.12](#))).

The root of the green domain tree is associated with ancestral gene *k12* in the *Fusarium* genome, showing that the green domain was not present in the least common ancestor of the gene family. Further, the root of the tree is a co-divergence with a duplication node in the gene tree, as can be seen from the red border on the blue root node. The inferred history also includes a domain transfer, represented by an edge highlighted in yellow. A yellow triangle, with a yellow ‘T’, appears halfway down the edge.

7. Click on the yellow arrow.

The names of the donor and recipient genes and species appear (Fig. 3.12), showing that a copy of the green domain, *green12*, was transferred from gene *k2* in *Fusarium* to gene *k3* in *Glomerella*.

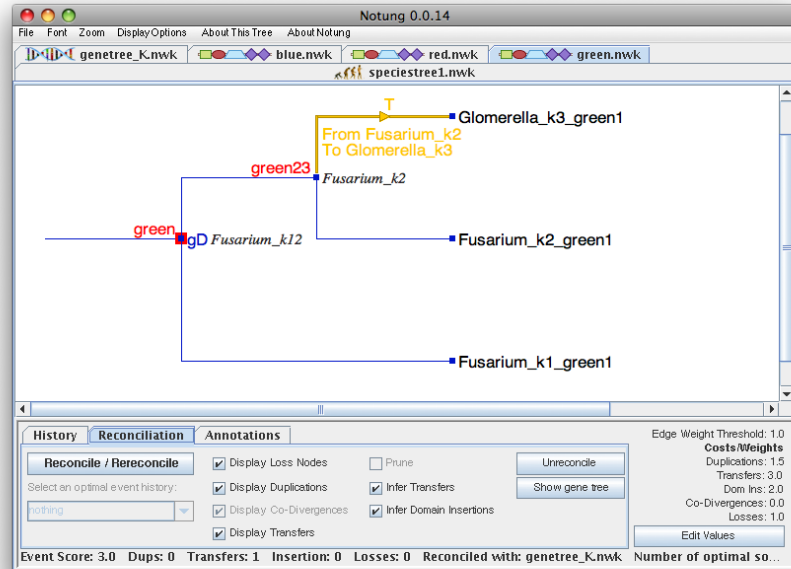


Figure 3.12: The reconciled tree for the green domain. The transfer event displays the donor and recipient genes and species. The gene and species associated with each internal domain node are also shown.

3.2 Example 2

The history of the multidomain family in Example 1 includes several types of domain events, specifically, domain duplications, insertions and transfers. It also provides an example of co-divergences with both speciation and duplication nodes in the gene tree. The history of the hypothetical multidomain family in a second example, shown in Fig. 3.13, demonstrates several events we have not yet encountered: domain losses, co-divergence with transfers in the gene tree, and how gene losses are handled in a domain tree reconciliation.

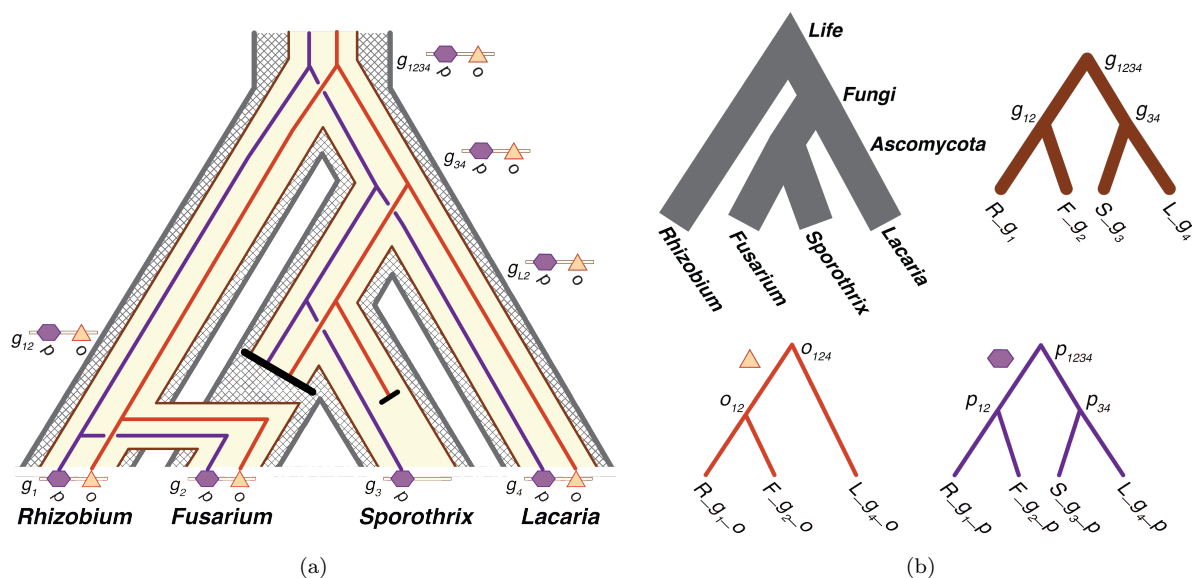


Figure 3.13: A second example of co-evolution on three different levels of organization: species evolution, gene family evolution, and domain shuffling. (a) Evolution of a hypothetical multidomain gene family in four genomes: Rhizobium (R), Fusarium (F), Sporothrix (S), and Laccaria (L). Domain trees (orange and purple) are embedded in the gene family tree (cream), which, in turn, is embedded in the species tree (grey). Colored boxes represent domains. Black lines indicate loss events. (b) The species, gene, and domain trees shown independently.

Hypothetical multidomain evolution, example 2. Fig. 3.13 shows the evolution of a second hypothetical multidomain family, gene family g . Family g has four present-day members, genes $g_1 - g_4$, and contains two constituent domains, represented by an orange triangle and a purple hexagon. The “true” history of this family is represented as a set of nested trees, with the domain trees embedded in the gene tree and the gene tree embedded in the species tree.

All four present-day species, *Rhizobium* (R), *Laccaria* (L), *Fusarium* (F), and *Sporothrix* (S), possess a single copy of this gene family. The common ancestor of the family is a single gene in the ancestor of all life. Although the presence of single copy in each species is suggestive of evolution by vertical descent, the member of the family in *Fusarium* was acquired by a horizontal gene transfer that replaced the ancestral copy of the gene. Note that a horizontal transfer that replaces the “native” copy is represented by a gene transfer and a gene loss in our model.

All four, present-day g genes possess a purple domain; only three present-day g genes possess an orange domain. Both the purple and orange domains were present in the common ancestor of the gene family, which in turn was present in the root of the species tree. Both domains were inherited with the locus (but not with the species) by vertical descent. Thus, *Fusarium* acquired the orange and purple domains via the same horizontal gene transfer that gave rise to the present day copy in *Fusarium* (F_g2). The purple domain has the same evolutionary history and tree topology as the gene family. The orange domain is absent from the present-day gene in *Sporothrix* (S_g3) due to a domain loss following the divergence of *Sporothrix* and *Fusarium*. The topology for the orange domain tree, therefore, agrees with the topology of the gene tree, except for the missing branch leading to gene S_g3 .

Input trees for this worked example are provided in the `Example_2` folder. As in Example 1, all reconciliations in Example 2 will be carried out with the default event costs. We begin by opening the species, gene and domain trees, and reconciling the gene tree.

Open the tree files

1. Click “**File → Open Species Tree**”
2. In the Open dialog box, select the `sampleTrees` folder, followed by the `Example_2` folder. Select the tree file `speciestree2.nwk` and click “**Open**”.

Once loaded, the species tree appears in the tree panel.

3. Click “**Display Options → Display Internal Node Names.**”

The names/identifiers of all internal nodes, i.e., ancestral species, should now appear on the species tree (Fig. 3.14(a)).

4. Click “**File → Open Gene Tree**” and open `genetree_G.nwk`.

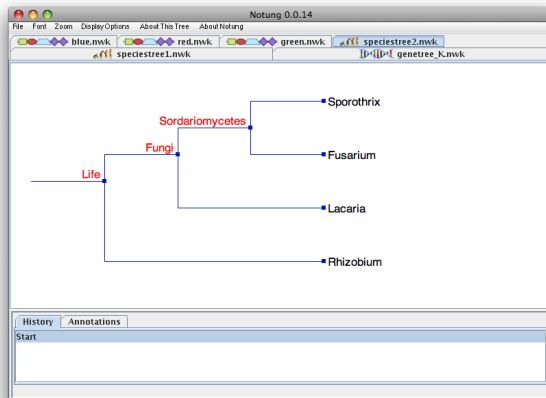
Once loaded, the gene tree is displayed in the tree panel.

5. Click “**Display Options → Display Internal Node Names.**”
The gene tree is now displayed with internal node names (Fig. 3.14(b)).
6. Click “**File → Open Domain Tree**”
7. In the Open dialog box, select the tree file and open `purple.nwk`.
Once loaded, the purple domain tree is displayed in the tree panel.
8. Click “**Display Options → Display Internal Node Names.**”
The purple domain tree is now displayed with internal node names (Fig. 3.14(c)).
9. Click “**File → Open Domain Tree**”
10. In the Open dialog box, select the tree file and open `orange.nwk`.
Once loaded, the orange domain tree is displayed in the tree panel.
11. Click “**Display Options → Display Internal Node Names.**”
The orange domain tree is now displayed with internal node names (Fig. 3.14(d)).

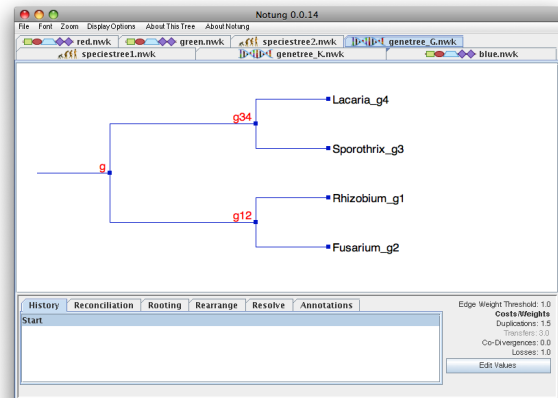
Reconciling the trees: Before the domain trees can be reconciled with the gene tree, that gene tree must first be reconciled with the species tree.

Reconcile the gene tree with the species tree

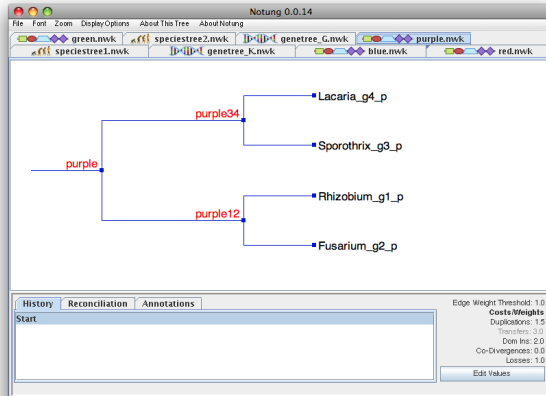
1. Click on the `genetree.G.nwk` tab to select the gene tree.
2. Click the “**Reconciliation**” tab in the task panel.
Recall that, by default, NOTUNG-DM reconciles gene trees with species trees using a DL event model. The “Infer Transfers” check box must be selected to add transfers to the event model.
3. Check the “**Infer Transfers**” box.
The text “Transfers: 3.0” in the parameter values area turns black, indicating that the DTL model will be used.
4. Click “**Reconcile/Rereconcile.**”
The **Reconciliation** dialog box appears, asking you to specify (1) the species tree and (2) the naming convention.



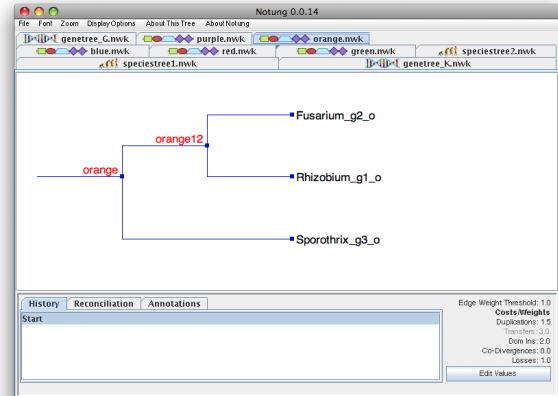
(a) Species tree



(b) Gene tree



(c) Purple Domain Tree



(d) Orange Domain Tree

Figure 3.14: Species, gene, and domain trees from Example 2, prior to reconciliation

5. Select `speciestree2.nwk` in the drop-down menu labeled “Please select a species tree to reconcile with.”

If you did not close³ the trees from the previous session, there will be two trees available for selection: `speciestree2.nwk` and `speciestree1.nwk`. You must be sure to select `speciestree2.nwk` here.

6. Under the section labeled “Specify Species Label” select “**Prefix of the gene label.**”

Recall that the prefix convention is required when the gene tree is being reconciled for later use with domain trees.

7. In the dialog box, click “**Reconcile.**”

³To close a tree: Select the tree to close; then select “Close” from the “File” menu.

The reconciled domain tree now appears in the tree panel. The Event Score of this reconciled tree is 4.0, the cost of one gene transfer and one gene loss.

- Click “**Display Options** → **Display Internal Species Node Names**” to view the names of the species associated with the internal gene nodes.

The reconciled gene tree (Fig. 3.15) now shows the inferred events and the species tree nodes associated with each internal gene tree node.

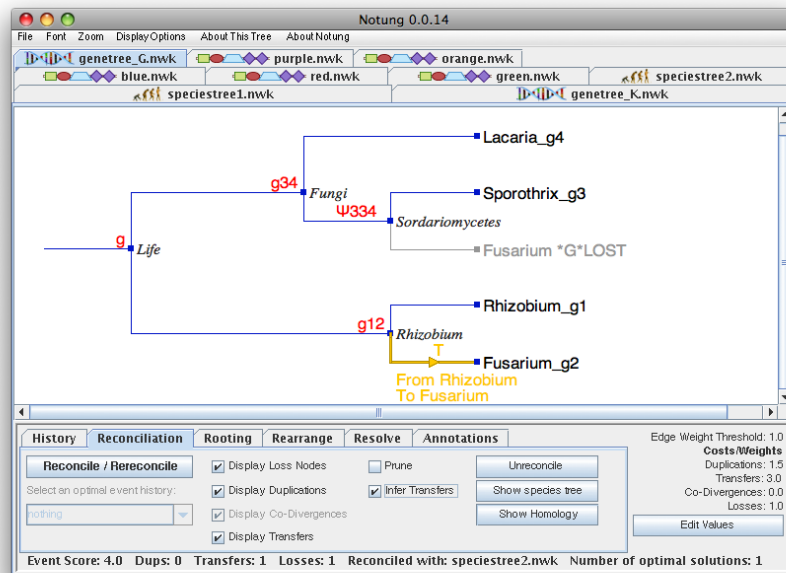


Figure 3.15: The reconciled gene tree displaying species associated with internal gene nodes. The transfer event is displaying the donor and recipient species.

Unlike the unreconciled gene tree in Fig. 3.14(b), the reconciled gene tree has five leaves and an extra internal node. This is because of the inferred gene loss in *Fusarium*; an additional leaf has been added representing the missing gene. This “loss node” is indicated by a grey node and the text “*Fusarium *G*LOST*,” which also appears in grey. This loss node is attached to the tree at the *pseudonode* $\psi334$. We refer to $\psi334$ as a *pseudonode* because it did not appear in the unreconciled gene tree. It represents the common ancestor of “*Fusarium *G*LOST*” and *Sporothrix_g3*. The markup shows that $\psi334$ is associated with the species tree node *Sordariomycetes*. This tells us that the loss occurred after the divergence at *Sordariomycetes* and before present day *Fusarium*. In other words, the branch from $\psi334$ to “*Fusarium *G*LOST*” shows the location of the gene in the gene tree had the loss not occurred.

In addition, the extant gene in *Fusarium*, *Fusarium_g2*, was the result of a gene transfer event, represented by an edge highlighted in yellow. A yellow triangle, with a yellow ‘T’, appears halfway down the edge.

9. Click on the yellow arrow.

The names of the donor and recipient species appear (Fig. 3.15), showing that the gene in *Fusarium* was acquired by a horizontal transfer from *Rhizobium*.

Reconcile the purple domain tree with the gene tree Now that the gene tree has been reconciled with the species tree, the domain trees can be reconciled with this reconciled gene tree to infer domain events.

1. Click on the `purple.nwk` tab to select the purple domain tree.
2. Click the “**Reconciliation**” tab.

Recall that, by default, NOTUNG-DM reconciles domain trees with gene trees using a DInL event model. The “Infer Transfers” check box must be selected to add domain transfers to the event model.

3. Check the “**Infer Transfers**” box.

The text “Transfers: 3.0” in the parameter values area turns black.

4. Click the “**Reconcile/Rereconcile**” button.

The **Reconciliation** dialog box appears, asking you to specify which gene tree to use for the reconciliation. Recall that only the prefix naming convention is supported for domain trees.

5. Select `genetree_G.nwk` in the drop-down menu labeled “Please select a gene tree to reconcile with.”

If you did not close the trees from the previous session, there will be two gene trees available for selection: `genetree_G.nwk` and `genetree_K.nwk`. You must be sure to select `genetree_G.nwk` here.

6. In the dialog box, click “**Reconcile**.”

The reconciled domain tree now appears in the tree panel.

7. Click “**Display Options** → **Display Internal Node Species Names**” to view the name of the genes and species associated with each internal domain node.

The reconciled domain tree now shows the gene and species tree nodes associated with each internal domain tree node (Fig. 3.16).

Gene events in the history of the gene family have contributed to the history of the purple domain and these can be seen in the markup of the reconciled domain tree. As in the previous example, several nodes are co-divergences with speciation nodes in the gene tree. Specifically, the co-divergences at the root and at node *purple34* are the result of speciation nodes in the gene tree, as represented by the small blue squares.

In addition, the reconciliation of the purple domain tree exemplifies co-evolution with two types of gene events that did not arise in Example 1: co-divergence with a gene transfer

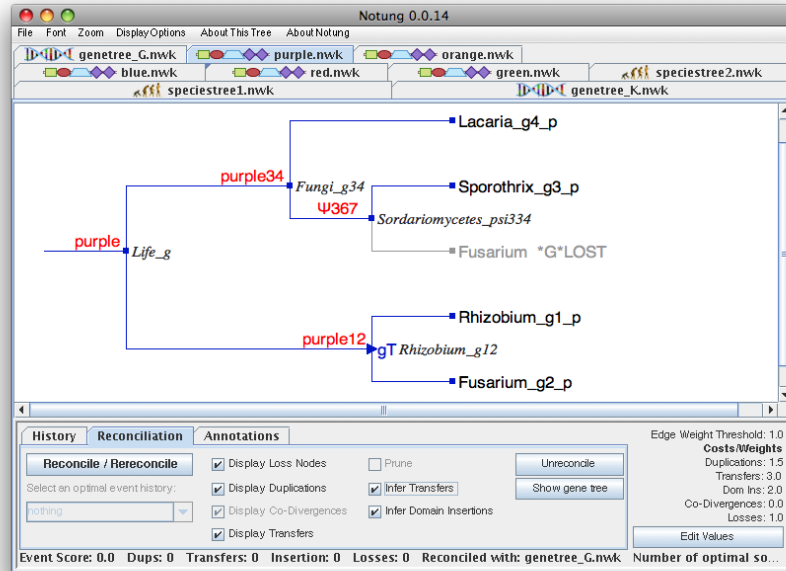


Figure 3.16: The reconciled purple domain tree displaying the gene and species associated with each internal domain node.

and with a gene loss. The co-divergence at node *purple12* is the result of a gene transfer from *Fusarium* to *Rhizobium*. In the markup, a co-divergence with a transfer event in the gene tree is displayed as a blue ‘gT’ next to a blue triangle at the node. Note that a co-divergence with a gene transfer is distinct from a domain transfer, which would have resulted in an additional copy of the purple domain in a different gene in a different species.

A co-divergence with a transfer (i.e., a triangle in the markup) is associated with the parent node, or donor, of the transfer in the gene tree. The markup in the domain tree does not explicitly indicate which domain tree node is associated with the recipient of the transfer. To determine this, it is necessary to look at the reconciled gene tree. Click on the `genetree.G.nwk` tab to select the reconciled gene tree. Inspection of the transfer edge, shown in yellow, shows that the recipient of the transfer is *Fusarium_g2*. Return to the purple domain tree by clicking on the `purple.nwk` tab. The leaf node associated *Fusarium_g2* corresponds to the transferred domain.

Domain tree reconciliation also explicitly accounts for domains that are missing due to a gene loss. In this example, the loss of a gene resulted in the loss of an instance of the purple domain. The text for the loss, “*Fusarium *G*LOST*,” indicates that the missing purple domain was the result of a loss of the whole gene, rather than the loss of a single domain. Again, a pseudonode ($\psi367$) appears to represent the ancestral purple domain that existed in gene $\psi334$ in *Sordariomycetes*. Note that it is only possible to infer that a domain instance is missing and that this absence is due to a gene loss because a reconciled gene tree is used in the inference process.

Reconcile the orange domain tree with the gene tree There are two causes of domain absence. A domain instance may be missing due to the loss of the gene that contained that domain instance or due to the loss of the domain. In the latter case, the domain is lost, but the “host” gene is retained.

The reconciliation of the purple domain tree, described above, exemplified the first case: a missing domain due to gene loss. Both cases arise in the history of orange domain: the loss of a gene that contained an orange domain and, in a separate event, the loss of an orange domain from a gene that was retained. To see how this plays out, we must first reconcile the orange domain tree.

1. Click on the **orange.nwk** tab to select the orange domain tree.
2. Click the **“Reconciliation”** tab and check the **“Infer Transfers”** box.
3. Click the **“Reconcile/Rereconcile”** button. In the dialog box, select **genetree_G.nwk** in the drop-down menu labeled **“Please select a gene tree to reconcile with.”**
4. In the dialog box, click **“Reconcile.”**

The reconciled domain tree now appears in the tree panel.

5. Click **“Display Options → Display Internal Node Names”** to view the internal node labels.

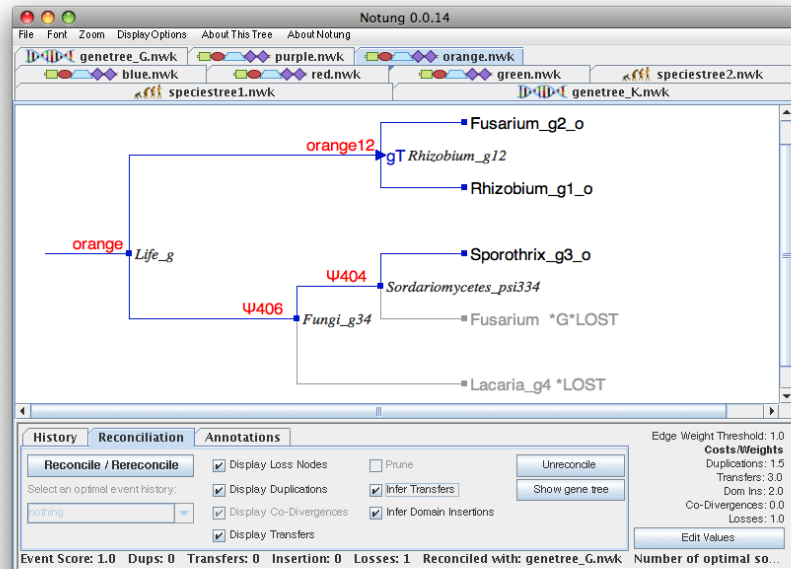


Figure 3.17: The reconciled tree for the orange domain, displaying internal node mappings.

The display (Fig. 3.17) now shows the reconciled domain tree with the name of the gene and species associated with each internal node of the domain tree. One domain loss was inferred in this reconciliation, resulting in an Event Score of 1.0. In the inferred history, all nodes in the domain tree arose through co-divergence events, which have zero cost. As in the purple domain tree, the co-divergence at the root is the result of a speciation event in the gene tree, while node *orange12* is a co-divergence with a transfer in the gene tree.

Comparison of the unreconciled and reconciled domain trees shows that two leaf nodes representing missing domains have been added, one corresponding to a gene loss and the other corresponding to a domain loss. In addition, pseudonodes have been inserted into the branch from *orange* to *Sporothrix_g3.o* to provide attachment points for these two loss nodes. The gene loss is designated by the text “**Fusarium *G*LOST**”, which indicates that a domain instance is missing due to the loss of a gene in *Fusarium*. The parent of this node, $\psi404$, is associated with the gene tree pseudonode $\psi334$ and the species tree node, **Sordariomycetes**.

The domain loss is designated “**Lacaria_g4 *LOST**”, which indicates the loss of an orange domain from gene *g4* in *Lacaria*. The parent of this node, $\psi406$, is associated with gene node *g34*. This indicates that the orange domain was present in the ancestral gene *g34*, but absent in the present day gene, *g4*. The species associated with $\psi406$ is **Fungi**, indicating that the domain loss occurred on the branch leading from **Fungi** to **Lacaria**.

This example illustrates the nomenclature for gene and domain losses in a reconciled domain tree. Gene losses are indicated by the text “***G*LOST**” preceded by a species name. The parent of a gene loss node is a pseudonode in the domain tree and is always associated with a pseudonode in the gene tree. Domain losses are indicated by the text “***LOST**” preceded by the names of the gene lacking this domain and its associated species. The parent of domain loss is pseudonode in the domain tree. The associated gene tree node is usually a true gene tree node, but can be a pseudonode under some circumstances.

3.3 Output

Domain event histories are presented graphically in the NOTUNG-DM GUI for exploratory analysis. The results of analyses performed with NOTUNG-DM can also be saved in various formats.

- The reconciled domain tree can be saved in Notung format, allowing the user to revisit the reconciled tree, with associated contextual information, at a later time. The reconciled domain tree can also be saved in other tree formats with some loss of information.
- A digital image of the reconciled tree can be saved in Portable Network Graphics (PNG) format.
- A summary of the general properties of the reconciled tree, such as the number of nodes and height, can be saved in a textual report. Note that the properties of the reconciled tree and the original tree may differ due to the insertion of loss nodes during reconciliation.
- Textual reports that summarize the inferred events and where they occurred in the domain, gene, and species trees can be generated in two formats:
 - The Event Summary report presents information in a “pretty print” format for easy reading.
 - The Parsible Statistics report presents the same information in a format that is suitable for scripting.

This section focuses on output file formats for reconciled domain trees. It is important to note that the details of the NOTUNG tree file format and the various textual reports differ depending on the type of tree (domain, gene or species) and whether or not the tree has been reconciled. The output files that can be generated for each type of tree are summarized in [Table 3.4](#). Furthermore, the information obtained is relevant only to (1) the tree currently selected and (2) the event history currently displayed. For example, if several domain trees were reconciled with the same gene tree, as was the case for both examples in this chapter, each reconciled domain tree must be saved separately.

3.3.1 Output Trees

Domain trees can be saved under three different formats:

Newick (.nwk) is a general and widely-used format for representing an evolutionary tree in plain text. The topology and branch lengths of a tree can be encoded in Newick format, but most inferred events cannot. Only loss events can be saved.

New Hampshire Extended (.nhx) is an extension of Newick that stores additional information in NHX comment fields. For each node, it is able to indicate whether a

Report	Domain Tree	Gene Tree	Species Tree
NOTUNG-2.8 Tree	Sec. 3.3.1	Appendix A.3	
Tree Statistics	Sec. 3.3.3	Sec. 3.4	
Event Summary	Sec. 3.3.4	Chap. 5, pg 55	×
Parsable		Chap. 5, pg 56	×

Table 3.4: NOTUNG-DM output file generation available based on tree type. For the gene and species trees, please see the listed sections in the Notung-2.8 Manual.

duplication event occurred, as well as the species associated with the node. It cannot store information on transfers and insertions.

Notung (.ntg) is a further extension of NHX that stores additional features in comment fields. The reconciled gene tree, the species tree, and the parameter values used in the reconciliation, as well as any color annotations, are also stored with the reconciled tree in Notung format (see [\(Fig. 3.19\)](#)).

Of the three formats, only Notung format can encode a complete description of an inferred event history.

Since the additional information encoded in both the NHX and Notung formats is stored in comment fields, in theory, any program that accepts input trees in the Newick or NHX formats should be able to read a reconciled tree saved in Notung format. Unfortunately, this is not always true, in practice. If you only save a reconciled tree in Newick, you will lose all the information on the event history, except for inferred losses. If you plan additional analyses using other phylogenetic software tools, you may find it helpful to save your reconciled tree in two, or even in all three, formats.

Save the reconciled domain trees.

1. Click on the `red.nwk` tab to select the reconciled red domain tree.
2. Click “**File** → **Save As**”

A dialog box appears, as seen in [Fig. 3.18](#). In this dialog box, NOTUNG-DM asks you to specify (1) the format in which to save the tree, (2) the name to use for the saved tree, and (3) where to save the tree.

3. In the Save dialog box, activate the drop down menu for “**Files of Type**” and select **Notung File Format**. Enter a descriptive file name, e.g. `redReconciled` and click “**Save**”.

The reconciled tree for the red domain is now saved in the file `redReconciled.ntg`. Note that the extension “.ntg” is automatically appended to the file name. The contents of this file are shown in [Fig. 3.19](#).

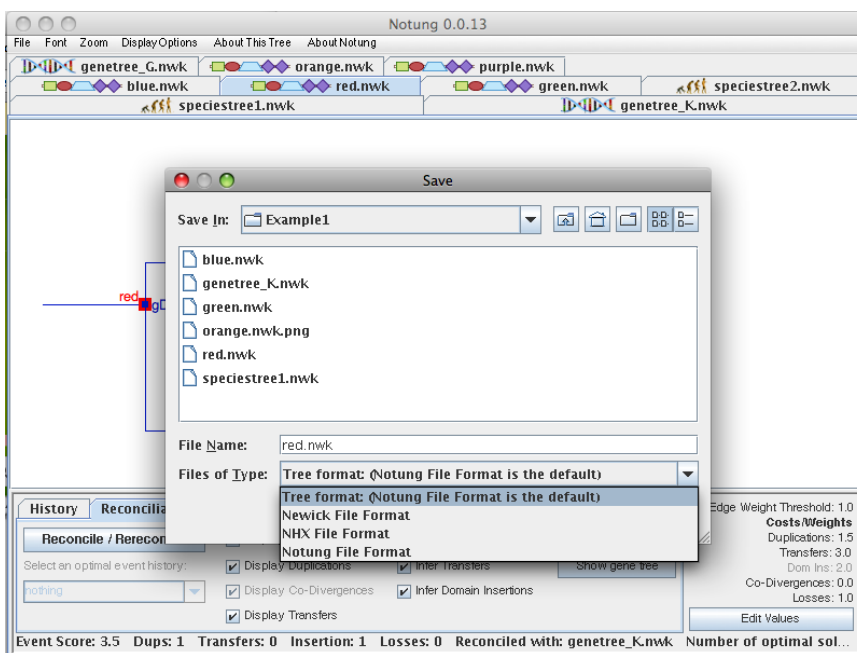


Figure 3.18: The Save dialog box with the Files of Type drop-down menu.

4. Close the tree by clicking “**File** → **Close**.”

These steps must be repeated for each reconciled tree you wish to save. For example, to save a complete record of Example 1, it is necessary to also save the reconciled blue and green domain trees, as follows:

5. Click on the `blue.nwk` tab to select the reconciled blue domain tree.
6. Click “**File** → **Save As**”
7. In the Save dialog box, select “**Files of Type** → **Notung File Format**”. Enter a descriptive file name, e.g. `blueReconciled` and click “**Save**”.

The reconciled tree for the blue domain is now saved in the file `blueReconciled.ntg`.

8. Close the tree by clicking “**File** → **Close**.”
9. Click on the `green.nwk` tab to select the reconciled green domain tree.
10. Click “**File** → **Save As**”
11. In the Save dialog box, select “**Files of Type** → **Notung File Format**”. Enter a descriptive file name, e.g. `greenReconciled` and click “**Save**”.

The reconciled tree for the green domain is now saved in the file `greenReconciled.ntg`.

12. Close the tree by clicking “**File** → **Close**.”

Reconciled red domain tree:

```
(
  (
    (
      Glomerella_k4_red2[&&NHX:G=Glomerella_k4:S=Glomerella],
      Glomerella_k4_red1[&&NHX:G=Glomerella_k4:S=Glomerella]
    )red44[&&NHX:G=Glomerella_k4:S=Glomerella:D=Y],
    Glomerella_k3_red2[&&NHX:G=Glomerella_k3:S=Glomerella:I=Y@Glomerella_k4@Glomerella_k3]
  )red344[&&NHX:G=Glomerella_k4:S=Glomerella],
  Glomerella_k3_red1[&&NHX:G=Glomerella_k3:S=Glomerella]
)red[&&NHX:G=k34:S=Glomerella:gD=Y];
```

Reconciled gene tree:

```
[&&NOTUNG-GENE-TREE
  (
    (
      Glomerella_k4[&&NHX:S=Glomerella],
      Glomerella_k3[&&NHX:S=Glomerella]
    )k34[&&NHX:S=Glomerella:D=Y],
    (
      Fusarium_k2[&&NHX:S=Fusarium],
      Fusarium_k1[&&NHX:S=Fusarium]
    )k12[&&NHX:S=Fusarium:D=Y]
  )k[&&NHX:S=Fungi]
]
```

Species tree:

```
[&&NOTUNG-SPECIES-TREE
  (Glomerella,Fusarium)Fungi
]
```

Parameters for reconciled gene tree:

```
[&&NOTUNG-EMB-PARAMETERS
  :T=1.0:INFERTTRANSFERS=Y
  :VERSION=3.0:SPECIESTAG=PRE
  :CD=1.5:CT=3.0:CL=1.0
]
```

Parameters for reconciled domain tree:

```
[&&NOTUNG-PARAMETERS
  :T=1.0:INFERTTRANSFERS=Y
  :INFERTINSERTIONS=Y:VERSION=3.0:SPECIESTAG=PRE
  :CD=1.5:CT=3.0:CI=2.0:CL=1.0
]
```

Figure 3.19: File `redReconciled.ntg`, the reconciled red domain tree in Notung format. Only text in the `teletype` format actually appears in the file; comments describing the sub-parts of the file appear in normal font. Indicators in the file also distinguish the entities stored in square brackets: `&&NOTUNG-GENE-TREE` - the reconciled gene tree; `&&NOTUNG-SPECIES-TREE` - the species tree; `&&NOTUNG-EMB-PARAMETERS` - parameters used during the gene tree-species tree reconciliation; `&&NOTUNG-PARAMETERS` - parameters used during the domain tree-gene tree reconciliation.

13. Click on the `genetree_K.nwk` tab to select the reconciled gene tree.
14. Click “**File** → **Save As**”
15. In the Save dialog box, select “**Files of Type** → Notung File Format”. Enter a descriptive file name, e.g. `genetree_K-Reconciled` and click “**Save**”.

The reconciled gene tree is now saved in the file `genetree_K-Reconciled.ntg`.

16. Close the tree by clicking “**File** → **Close**.”

Species trees play a passive role in NOTUNG-DM analyses; gene and/or domain tree reconciliation do not modify the species tree. Therefore, there is no need to save the species tree in Example1⁴.

17. Click on the `speciestree1.nwk` tab to select the species tree.

18. Close the tree by clicking “**File** → **Close**.”

When a reconciled domain tree in Notung format is reopened in NOTUNG-DM, the gene and species trees can be extracted.

To open an embedded, reconciled gene tree in a Notung format domain tree file:

1. Open the Notung format domain tree file.
2. Click the Reconciliation tab to enter reconciliation mode.
3. Click the “**Show Gene Tree**” button.

To open an embedded species tree in a Notung format gene tree file:

1. Open the Notung format gene tree file.
2. Click the Reconciliation tab to enter reconciliation mode.
3. Click the “**Show Species Tree**” button.

3.3.2 Saving Tree Images

In addition, a picture of the reconciled tree, as seen in the NOTUNG-DM GUI can be saved as a PNG image.

To save an image of the tree as currently displayed in the NOTUNG-DM GUI:

- Click “**File** → **Save Current View as Image (PNG)**.”

NOTE: This option saves only the image currently visible in the tree panel. If you have zoomed in on a tree, the PNG will save only the section in view.

To save an image of the whole tree as a PNG file:

- Click “**File** → **Save Whole Tree as Image (PNG)**.”

NOTE: This option saves an image of the entire tree.

⁴More generally, there is usually no reason to save a species tree in NOTUNG-DM. One exception is when color annotations have been added to the species tree. Color annotation is a standard feature in NOTUNG-2.8 and is described in Notung-2.8 Manual Chap. 10.

3.3.3 General Tree Statistics

NOTUNG-DM reports information on tree characteristics from the General Tree Statistics pop-up box under the “**About This Tree**” menu. The General Tree Statistics report for gene and species trees is described in Notung-2.8 Manual Sec. 3.4. The report for reconciled domain trees is described here. (See [Fig. 3.20](#) for an example.)

To get general statistics for a tree:

- Click “**About This Tree** → **General Tree Statistics**.”

A window will appear containing a summary of properties of the tree that is currently displayed. To copy this information into your favorite text editor, click the “Copy to Clipboard” button, and paste in the text editor.

The first line, **Statistics for:** `file_name`, states the name of the tree file. If the tree has been reconciled, a summary of the number of inferred events is provided under the heading **Reconciliation Information**:

Duplications: The total number of domain duplications.

Transfers: The total number of domain transfers.

Insertions: The total number of domain insertions.

Losses: The total number of domain losses. Note that domains that are missing due to a gene loss are not included in this quantity.

Co-Divergences: The total number of co-divergences, including speciations, gene duplications, and gene transfers.

The next section of this report provides statistics on the topology of the reconciled tree with and without loss nodes, under the headings **Tree Without Losses** and **Tree With Losses**.

Total nodes: The total number of nodes in the domain tree.

Internal nodes: The total number of internal nodes (**Total nodes** minus **Leaf nodes**).

Leaf nodes: The total number of leaves in the domain tree.

Polytomies: The total number of polytomies in the tree. This number will be 0 for reconciled domain trees, since only binary domain trees can be reconciled.

Size of largest polytomy: The number of children of the largest polytomy in the tree. This number will be 0 for reconciled domain trees.

Height: The length of the longest path from a leaf node to the root.

Edge Weight Range: The range of edge weights in the tree in the form [minimum edge weight, maximum edge weight]. This line will not appear if the tree has no edge weights.

These sections are then followed by the analogous information for the reconciled gene tree, with which the domain tree was reconciled. The last section reports the characteristics of the associated species tree.


```
Statistics for: orange.nwk
Reconciliation Information
- Duplications: 0
- Co-Divergences: 0
- Transfers: 0
- Losses: 1
- Number of Temporally Feasible Optimal Solutions: 1

Tree Without Losses
- Total nodes: 7
- Internal nodes: 4
- Leaf nodes: 3
- Polytomies: 0
- Size of largest polytomy: 0
- Height: 3

Tree With Losses
- Total nodes: 9
- Internal nodes: 4
- Leaf nodes: 5
- Size of largest polytomy: 0
- Height: 3

This domain tree is reconciled.
- In GUI mode, the reconciled gene tree can be viewed
  by clicking "Show Gene Tree" in the Reconcile Panel.

Statistics for: geneTree_G.nwk
Reconciliation Information
- Duplications: 0
- Co-Divergences: 0
- Transfers: 0
- Losses: 0
- Number of Temporally Feasible Optimal Solutions: unknown

Tree Without Losses
- Total nodes: 8
- Internal nodes: 4
- Leaf nodes: 4
- Polytomies: 0
- Size of largest polytomy: 0
- Height: 3

Tree With Losses
- Total nodes: 9
- Internal nodes: 4
- Leaf nodes: 5
- Size of largest polytomy: 0
- Height: 3

This gene tree is reconciled.
- In GUI mode, the pruned species tree can be viewed by
  clicking "Show Species Tree" in the Reconcile Panel.
- From the command line, the pruned species tree can
  be saved with the option --stpruned.

Statistics for species tree pruned from: speciesTree.nwk
- Total nodes: 7
- Internal nodes: 3
- Leaf nodes: 4
- Polytomies: 0
- Size of largest polytomy: 0
- Height: 3
```

Figure 3.20: Tree statistics for the orange domain

3.3.4 Event Summary and Parsable Statistics

Two types of reports are available, the Event Summary and Parsable Statistics. Similar event information is contained in both reports; the Parsable Statistics file contains some additional information on tree size. The Event Summary report is in a “pretty print” format that is easy to read. The Parsable Statistics report is designed for downstream analysis using a scripting language. For reconciled domain trees, the information provided in each of these reports is described in detail below. Similar reports can be obtained for reconciled gene trees, detailed in Notung-2.8 Manual Sec. 5.4.

To view event details in a human readable format:

- From the “**About This Tree**” menu, above the tree panel, select the “**Event Summary**” options. A new window displaying information will appear. This option is grayed out if the tree has not been reconciled.

A window will appear containing a “pretty print” summary of events inferred on the tree that is currently displayed. To copy this information into your favorite text editor, click the “Copy to Clipboard” button, and paste in the text editor.

To view event details in an easily parsable format:

- Select the “**Parsable Statistics**” item from the “**About This Tree**” menu above the tree panel. A new window displaying information will appear. This option is grayed out if the tree has not been reconciled.

A window will appear containing a tab-delimited summary of events inferred on the tree that is currently displayed. To copy this information into your favorite text editor, click the “Copy to Clipboard” button, and paste in the text editor.

Event Summary

The contents of the Event Summary report are organized into tables, summarizing the inferred duplications, transfers, insertions, co-divergences, and losses. These tables include information on the timing of domain duplications (upper and lower bounds on the time of duplication), domain insertion and transfer (the gene and species from which the domain originated and the gene and species which received the domain), and domain losses (the gene and species in which the loss occurred), as well as timing of co-divergences with the gene tree. The duplication, insertion, and transfer tables each contain one line per event. Two tables are provided for losses, summarizing the number of losses in each gene and the number of losses in each species, respectively. The association between co-divergences in the domain tree and nodes in the gene and species tree are tabulated in the co-divergence table. The last table summarizes the total number of events of each type that are associated with each node

in the gene tree. If no event of the specified type was inferred in the current reconciliation, the corresponding table is left empty. These tables are preceded by the Event Score of the reconciled tree. The report closes with five lines summarizing the total number of inferred events in each category.

Duplication Table: Inferred duplications are summarized in a table with five columns. The domain tree node representing the duplication is listed in the first column. The remaining four columns describe the location of this event in the gene and species trees. Columns 2 and 3 contain the lower and upper bounds on the event in the gene tree, expressed as node names in the gene tree. Lower and upper bounds relative to the species tree are given in Columns 4 and 5. The total number of duplications appears immediately following this table.

Duplications				
Domain	L. Bound Gene	U. Bound Gene	L. Bound Species	U. Bound Species
red44	Glomerella_k4	Glomerella_k4	Glomerella	Glomerella
Duplications: 1				

Figure 3.21: Duplication event summary for the red domain in Fig. 2.2.

Insertion Table: For each insertion, the first two columns list the domain tree nodes corresponding to the parent and child (i.e., donor and recipient) of the insertion. Columns 3 and 4 give the donor and recipient genes. Column 5 indicates the species, expressed as a species tree node, in which the insertion occurred. (Recall that insertions are only allowed between genes in the same species.) The total number of insertions is listed at the end of this table. If insertions are not allowed in the event model, this table will not appear in the report.

Insertions				
From(domain)	To(domain)	From(gene)	To(gene)	Species
red344	Glomerella_k3_red2	Glomerella_k4	Glomerella_k3	Glomerella
Insertions: 1				

Figure 3.22: Insertion event summary for the red domain in Fig. 2.2.

Transfer Table: For each transfer, the first two columns list the nodes in the domain tree where the transfer originated and terminated. The timing of the transfer event, relative to the gene and species trees, is given in the subsequent four columns. Columns 3 and 4 report the donor and recipient genes; Columns 5 and 6 report the donor and recipient species. The total number of transfers appears at the end of this table. Note that if transfers are not included in the event model, this table will not appear.

Transfers					
From(domain)	To(domain)	From(gene)	To(gene)	From(species)	To(species)
green23	Glomerella_k3_green1	Fusarium_k2	Glomerella_k3	Fusarium	Glomerella
Transfers: 1					

Figure 3.23: Transfer event summary for the green domain in Fig. 2.2.

Loss Table: Domain losses are summarized in two tables, according to the gene and species taxa in which they occurred. The first table summarizes domain losses per gene. This table contains one line for each node in the gene tree. The name of the gene tree node appears in Column 1. The species tree node associated with this gene is given in Column 2. Column 3 gives the total number of inferred domain losses sustained by this gene.

The second table summarizes domain losses per species. Column 1 lists each node in the species tree, and Column 2 lists the number of domain losses associated with that species.

The total number of losses is given immediately following these tables. Note that these tables include domain losses; domains missing due to gene losses are not reported here.

Losses		
Gene	Species	No. of Losses
Fusarium_g2	Fusarium	0
Sporothrix_g3	Sporothrix	0
g12	Rhizobium	0
psi1087	Sordariomycetes	0
g34	Fungi	0
Rhizobium_g1	Rhizobium	0
Fusarium *G*LOST	Fusarium	0
Lacaria_g4	Lacaria	1
g	Life	0
Species	No. of Losses	
Life	0	
Sordariomycetes	0	
Rhizobium	0	
Sporothrix	0	
Lacaria	1	
Fusarium	0	
Fungi	0	
Losses: 1		

Figure 3.24: Loss summary for the orange domain in Fig. 3.13.

Co-divergence Table: This table contains one line for each co-divergence in the domain tree. The domain node corresponding to the co-divergence is listed in the first column. Columns 2 and 3 give the associated gene and species tree nodes, respectively. Column 4 provides the event that caused the divergence in the gene tree: either gene duplication, gene transfer, or speciation. The total number of co-divergences is given at the end of the table.

Co-divergences			
Domain	Gene	Species	Event Type
=====			
blue	k	Fungi	Speciation
blue12	k12	Fusarium	Gene Duplication
blue34	k34	Glomerella	Gene Duplication
Co-Divergences: 3			

Co-divergences			
Domain	Gene	Species	Event Type
=====			
purple	g	Life	Speciation
purple34	g34	Fungi	Speciation
purple12	g12	Rhizobium	Gene Transfer
Co-Divergences: 3			

Figure 3.25: Co-divergence summaries for the blue domain (top) in Fig. 2.2 and purple domain (bottom) in Fig. 3.13.

Event Counts Table: This table summarizes the events associated with each node in the gene tree in nine columns. Each line corresponds to a single node in that tree. The gene tree node and its associated species appear in Columns 1 and 2. Columns 3, 8, and 9 give the number of duplications, losses, and co-divergences associated with that gene, respectively. Column 4 gives the number of times an insertion originated from this gene. The number of insertions the gene received is given in Column 5. Similarly, the number of times a domain transfer originated from or was received by this gene is given in Columns 6 and 7, respectively. The total number of each type of event follows this.

Gene	Species	Dups	Ins_to	Ins_from	Trans_to	Trans_from	Losses	Co-div
=====								
Fusarium_k2	Fusarium	0	0	0	0	0	0	0
Fusarium_k1	Fusarium	0	0	0	0	0	0	0
k12	Fusarium	0	0	0	0	0	0	0
Glomerella_k3	Glomerella	0	1	0	0	0	0	0
Glomerella_k4	Glomerella	1	0	1	0	0	0	0
k	Fungi	0	0	0	0	0	0	0
k34	Glomerella	0	0	0	0	0	0	1
Duplications: 1								
Domain Insertions: 1								
Transfers: 0								
Losses: 0								
Co-divergences: 1								

Figure 3.26: Event count table for the red domain in Fig. 2.2.

Parsable Statistics

This file contains similar information to that in the Event Summary file, but in a format that is easily parsed in a scripting language. (See Fig. 3.27 for an example.) The first line is a tab-delimited set of values, with the header on the line below. The values, rather than the

header, are given in the first line in order to support scripts that process the first line in the file. The data included in this line are described in [Table 3.5](#).

item	description	item	description
Cost	DTIL score	minEW,	Minimum and maximum edge weights
nD	No. of duplications	maxEW	
nT	No. of transfers	Roots	NA (number of optimal roots)
nIn	No. of insertions	Cand	No. of candidate, optimal solutions
nCD	No. of co-divergences	Feas	No. of feasible, optimal solutions
nL	No. of losses	cD	Duplication cost
L(D)	No. of domain tree leaves	cT	Transfer cost
D	No. of domain tree nodes	cI	Insertion cost
G	No. of gene tree nodes	cCD	Co-divergence cost
S	No. of species tree nodes	cL	Loss cost

Table 3.5: Header for the data presented in the first line of the “**Parsable**” file.

The remainder of the file provides the same information as the Event Summary report, but in a tab-delimited format. Each line begins with #<x>, where <x> is a tag corresponding to one of the tables described in “**Event Summary**.” These tags, described in [Table 3.6](#), allow scripts to identify the type of event reported, on a line-by-line basis. They can also be used to `grep` the file to return only the event table of interest.

Note that co-divergences events in the Parsable Statistics file are reported as abbreviations. In other words, what appears as “speciation”, “gene duplication”, and “gene transfer” in the co-divergence table of the Event Summary file, are abbreviated ‘S’, ‘gD’, and ‘gT’, respectively in the co-divergence table of the Parsable Statistics file.

flag	Table
#D	Duplications
#I	Insertions
#T	Transfers
#LinG	Losses, by gene node
#LinS	Losses, by species node
#CoDiv	Co-divergences
#S	Event summary counts, by gene node

Table 3.6: Parsable file line identifier codes.

3.5	1	0	1	1	0	4	7	7	3	100,100	0	1	1	1.5	3.0	2.0	0.0	1.0	
Cost	nd	nT	nIn	nCD	nL	IL(D) D	D	G	S	minEW,maxEW	Roots	Cand	Feas	CD	cT	cI	cCD	cL	
#D	Domain Node	U. Bound Gene	L. Bound Gene	U. Bound Species	L. Bound Species														
#D	red44	Glomerella_k4	Glomerella_k4	Glomerella_k4	Glomerella														
#I	From(domain)	To(domain)																	
#I	red344	Glomerella_k3_red2																	
#T	From(domain)	To(domain)																	
#Ling	Gene	Species Losses																	
#Ling	k	Fungi	0																
#Ling	k12	Fusarium	0																
#Ling	Glomerella_k4	Glomerella	0																
#Ling	Fusarium_k1	Fusarium	0																
#Ling	Glomerella_k3	Glomerella	0																
#Ling	k34	Glomerella	0																
#Ling	Fusarium_k2	Fusarium	0																
#Lins	Species Losses																		
#Lins	Fungi	0																	
#Lins	Glomerella	0																	
#Lins	Fusarium	0																	
#Codiv	Domain Gene	Species Event																	
#Codiv	red	k34	Glomerella	gd															
#S	Gene	Species Dups	Ins_from																
#S	k	Fungi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#S	k12	Fusarium	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#S	Glomerella_k4	Glomerella	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#S	Fusarium_k1	Fusarium	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
#S	Glomerella_k3	Glomerella	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
#S	k34	Glomerella	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
#S	Fusarium_k2	Fusarium	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.27: Text reported for the Parsable file for the red domain in Fig. 2.2.

3.4 Potential Pitfalls

In this guide, we have introduced the basic functions of NOTUNG-DM using two hypothetical multidomain families as examples. The histories of these families exemplify all of the domain events in our multidomain model, as well as co-divergences with various types of gene events. These examples demonstrate the inferential power of reconciliation with three levels of organization (domain, gene, and species). This is in contrast to current approaches that treat domains as characters and infer domain gain and loss by applying parsimony methods to obtain infer ancestral character states. For example, NOTUNG-DM’s reconciliation-based approach correctly reconstructs the history of the domain events in the family in Example 1. In [Section 2.2.2](#), we saw that Wagner parsimony could not correctly reconstruct the history of events in this family.

Here, we discuss challenges and pitfalls that may arise with more complex situations.

Composite multidomain histories NOTUNG-DM reconstructs the history of events in a single domain family, relative to the associated gene and species trees. Given a family with domains from two or more domain families, it is possible to reconstruct all of the domain events in the history of the multidomain gene family, by reconciling each of the domain trees separately. In Example 1, we reconstructed the history of the blue, red, and green domains in three different reconciliations. Similarly, two separate reconciliations were required to infer the events in the orange and purple domain families in Example 2.

At present, NOTUNG-DM is not able merge multiple domain event histories to reconstruct the history of the multidomain family as a whole. Nor does it consider the N- to C-terminal ordering of domains. However, the user can combine the histories of individual domains in a post-processing step. This process, described formally in [Chapter 4](#), results in a composite history, which shows the ancestral domain architectures and the evolution of individual domains in relation to one another in a gene tree.

Event model: In NOTUNG-DM, the user can select the set of events that will be used for reconciliation by checking or unchecking the appropriate box in the reconciliation task panel. In NOTUNG-DM’s domain event model, there are two kinds of horizontal events: domain insertions, in which a gene acquires a domain from another gene in the same genome, and domain transfers, in which a gene acquires a domain from another gene in a different genome. By default, domain insertions are selected in NOTUNG-DM’s reconciliation mode, while transfers are turned off for both gene and domain reconciliation. In the worked examples in previous sections, the “Infer Transfers” function was selected to provide a broad demonstration of the capabilities of the software. This should not be taken as a universal recommendation for a model with transfers.

There are several points to consider when deciding whether to include transfers in the event model. First, transfers only make sense in species for which there is evidence to suggest that horizontal transfers occur. Complex multidomain families are particularly evident in metazoa, species for which there is little evidence of horizontal transfer. For analysis of these

families, a model without horizontal transfers may be more appropriate.

Second, the gene event and domain event models in NOTUNG-DM are uncoupled: There is nothing to prevent you from using a model with transfers when reconciling the gene tree and a model without transfers when reconciling the domain tree, or vice versa. That said, it is not clear under what circumstances it would be realistic to assume the occurrence of horizontal gene transfers, but not horizontal domain transfers (or the reverse). Generally, we suggest that transfers be included in both models or in neither model.

Finally, reconciliation with transfers entails various complications, which are best avoided unless the underlying biology demands them. Two of these complications, multiple optimal solutions and temporal feasibility, are discussed further below. In addition, the increased computational complexity of reconciliation with transfers leads to longer running times, especially with large trees.

Event costs: The worked examples in this chapter used default event costs. Different costs can result in different inferred histories. For some data sets, you may wish to use different event costs. You may also find it useful to analyze the family with several different costs to investigate the sensitivity of the inferred history to parameter choice. For instructions on how to change parameters in NOTUNG-DM, see Notung-2.8 Manual Sec. 3.5.

Unfortunately, at this time, there is no theoretical basis for selecting appropriate costs for a given data set. However, various rules of thumb can be helpful in deciding which alternate costs to try. In reconciliation with horizontal events (transfers or insertions), scenarios frequently arise in which a transfer replaces a duplication and one or more losses. An example of this is given in [Fig. 2.1](#), in which the disagreement between the gene and species trees can be explained by a duplication and three losses or by a transfer and one loss. The history with one transfer and one loss is preferred whenever the cost of a transfer is lower than the cost of a duplication and two losses. Scenarios where a transfer can replace several losses are also fairly common. With this in mind, one way to think about choosing costs is to consider the relative plausibility of a history with many losses versus a history with a transfer. In fact, the default costs were chosen so that the cost of a horizontal event is greater than the cost of a duplication and a loss, but less than the cost of a duplication and two losses.

Another consideration is that the problem becomes more complex as the number of inferred transfers increases. The running time, the number of degenerate solutions, and the probability that there will be no feasible solution, all tend to increase with the number of transfers. These problems can be ameliorated by increasing the transfer cost to reduce the number of transfers that are inferred. Alternatively, lowering the loss cost can also reduce the number of inferred transfers.

Multiple optimal reconciliations: Both hypothetical examples in this chapter had a single, temporally feasible solution. However, more complicated and potentially problematic behavior may arise with more complex multidomain families. Two particular challenges are degeneracy (the occurrence of more than one optimal history) and temporal infeasibility, which are discussed in [Chapter 2](#) of this manual. More information is available in Notung-2.8

Manual Sec. 5.2.1 and Notung-2.8 Manual Sec. 5.2.2, respectively.

If there is more than one feasible event history with the same total reconciliation score, the status bar will display the number of feasible optimal solutions. A green circle over a node in the domain tree indicates the existence of multiple event histories for the subtree rooted at that node. The user can graphically browse through all optimal event histories by clicking on these green circles. If your analysis generates two or more optimal reconciliations, in order to save all of them, you must select each history in turn, save it, and go on to the next one. Similarly, the reports obtained via the **About This Tree** menu only describe the history that is currently selected. In order to save the event summary or parsable statistics for all optimal solutions, you must visit each one.

One of the challenges associated with degeneracy is the problem of interpreting a collection of “equally good” hypotheses. Increasing the transfer cost or reducing the lost cost may reduce the number of possibilities that must be considered. Additionally, there are various ways of combining information from several inferred histories, for example, by weighting events based on the number of histories in which they are observed. An example of this is described in [19].

Temporal infeasibility: NOTUNG-DM generates candidate event histories and then checks each one to ensure that it is temporally feasible. A history is temporally feasible if it is possible to assign a time-stamp to the donor and recipient of each insertion and transfer, without traveling backward in time. Only temporally feasible event histories are reported to the user.

If the number of candidate histories is very large, then testing temporal feasibility of all candidates can result in very long running times. If all candidate solutions are temporally infeasible, a dialog box with a warning will pop up, and no solution will be displayed in the tree panel. If this occurs, you may be able to obtain a solution by increasing the transfer cost and/or decreasing the loss cost.

Failure to find a feasible solution will impact the information that can be obtained from the **About This Tree** menu. If a domain tree has no feasible solution, “**About This Tree** → **General Tree Statistics**” will only display statistics related to the unreconciled tree. Similarly, if a domain tree has no feasible solution, “**About This Tree** → **Event Summary**” and “**About This Tree** → **Parsable Statistics**” will result in an error message, since there are no inferred events to summarize.

Chapter 4

Reconstructing ancestral domain architectures

In the previous steps, we inferred the history of each domain family separately. The current version of NOTUNG-DM does not combine these individual domain histories to infer ancestral domain architectures. However, the ancestral domain architectures can be reconstructed from NOTUNG-DM output in a post-processing step. Here, we demonstrate, in terms of the reconciled gene tree, how this information can be used to map domain events onto the gene tree and to infer ancestral domain architectures. Users should note that while Notung provides all the information required to perform this reconstruction, this process is not currently automated. This chapter is more technical than [Chapter 3](#). The reconstruction procedure is illustrated for Example 1 in [Section 4.3](#) as a quick and intuitive guide to this process.

4.1 Input

To generate the reconstruction, we need to know the association between ancestral nodes in the domain, gene, and species trees. In addition, we need to know the inferred events, as well as the timing of those events relative to the gene tree. For this procedure, loss nodes must be displayed in both the reconciled gene tree and the reconciled domain trees. Pseudonodes, i.e., nodes connecting a lost leaf to a branch, will also be displayed. Such nodes are considered to be the result of a co-divergence event followed by a loss in one child. All this information can be obtained from the reconciled domain and gene trees in the NOTUNG-DM GUI.

To view the gene and species associated with each internal node in the domain tree, selecting “**Display Options** → **Display Internal Node Species Names**” when viewing the reconciled domain tree. Recall that the association is formatted as `Species_GeneID` and that these identifiers may refer to internal nodes in the gene or species tree. The species mapping for internal gene nodes can be viewed by selecting “**Display Options** → **Display Internal Node Species Names**” when viewing the reconciled gene tree.

To view the internal node names for these trees, the user may select “**Display Options**

→ **Display Internal Node Names**” when viewing the gene or species trees.

The set of events and their timing relative to nodes in the gene tree may also be obtained from the event file, which is generated by selecting “**Event Summary**” from the “**About This Tree**” menu. (See [Section 3.3.4 - Event Summary](#) on page 50 for more information.) However, using the image of a reconciled tree, either from the GUI or a saved PNG, is more intuitive; thus we explain the reconstruction here, with a focus on using the image as input.

4.2 The Reconstruction Process

Given a reconciled gene tree with losses, T_{GS}^* , and a set of reconciled domain trees with losses for each domain D_i , $T_{D_iG}^* \in \{T_{D_1G}^*, T_{D_2G}^*, \dots, T_{D_nG}^*\}$, the following procedure can be used to construct a composite gene tree history with ancestral architectures and inferred events from all domains. This history is constructed by transferring information from each reconciled domain tree to the appropriate branch in the reconciled gene tree. The ancestral architectures are determined by the mapping between domain tree nodes and gene tree nodes, with counts of domains adjusted for the inferred events. Formally, we define the problem of reconstructing a domain event history as follows:

Domain Shuffling Reconstruction (DSR)

Domain events: $\{\mathcal{O}, \mathcal{C}_S, \mathcal{C}_D, \mathcal{C}_T, \mathcal{D}, \mathcal{T}, \mathcal{I}, \mathcal{L}\}$.

Input:

T_{GS}^* : A rooted, binary, *DTL*-reconciled gene tree with stubs representing lost genes.

$\{T_{D_1G}^*, T_{D_2G}^*, \dots, T_{D_nG}^*\}$: The set of rooted, binary, reconciled domain trees with losses, for each domain D_i .

Output:

T_{GS}^* with edges added to represent horizontal events. In addition, each node $g \in V_G^*$ is annotated, for each domain D_i , with

The number of instances of domain D_i in gene g .

The set of events that occurred in domain tree D_i along the branch leading to node g .

Domain Shuffling Reconstruction. For each domain tree $T_{D_iG}^*$ in $\{T_{D_1G}^*, T_{D_2G}^*, \dots, T_{D_nG}^*\}$, repeat the following procedure on the reconciled gene tree T_{GS}^* . Traverse the domain tree depth-first.

1. Initialize the number of instances of domain D_i in each gene $g \in T_{GS}^*$ to 0.
2. Look at the root of the domain tree, and determine when the domain first originated in the gene family¹.

¹Notung does not report domain origins, even if the domain is not present in the root of the gene family. This step infers the origination.

- (a) Let ρ_{D_i} be the root of the domain tree.
 - (b) Let g be the gene node to which ρ_{D_i} is mapped.
 - (c) This was the first appearance of the domain in the gene family. Set the number of domain instances at g to 1.
 - i. If the root of the domain tree does not map to the root of the gene tree (*i.e.*, $g \neq \rho_G$), then there was an origination event on the edge of the gene tree leading to g . Annotate that edge with an origination event \mathcal{O} .
 - ii. Else, the domain was present in the most recent common ancestor of the gene family.
3. Starting at the root of the domain tree, traverse the domain tree in pre-order by starting the following procedure with ρ_{D_i} and g .
4. For a given domain node d and its associated gene node g :
- (a) If the event at d is a *domain duplication*, the number of D_i domains in gene g increased by 1. Annotate the edge leading to g with a duplication in D_i
 - (b) If the event at d is a *co-divergence* (of any type), the domain was passed down to both children of g . Increment the number of D_i domains in each child of g .
 - (c) If the event at d is a *domain transfer* or *domain insertion*, we must consider which child of d was the domain that was transferred or inserted. Every transfer/insertion donor is an internal node and has two children, one of which is the recipient of the event. In the GUI, the recipient can be identified as the child connected to d by an edge highlighted in yellow for a domain transfer or in purple for a domain insertion. Let d' be this child. Node d represents the *donor* of the horizontal event, while node d' represents the *recipient*.
 Let g' be the gene that maps to d' . Then g represents the gene where the domain originated, while g' represents the gene that received the horizontally transmitted domain. In T_{GS}^* , add an arrow *from* the edge leading to g *to* the edge leading to g' .
 Increment the number of domains at g' by 1.
 - (d) If domain node d is a *domain loss* leaf, then there was one less copy of domain D_i in g . Annotate the edge leading to g with a loss, and decrement the number of domain instances at g by one.
 - (e) If d is not a leaf node, repeat Step 4 for the children of d .

NOTE: Formal pseudocode is provided in Alg. 4.1. This format may be useful to users who are comfortable with such formalizations.

4.3 Example 1

We illustrate this process with the reconciled domain trees from Example 1, considering each domain family in turn. Ancestral domain architectures are reconstructed by a depth first traversal of the gene tree. At each node in the gene tree, the number of domains is determined based on the number of nodes in the domain tree that are associated with that gene node, adjusting for the events that occurred in that gene.

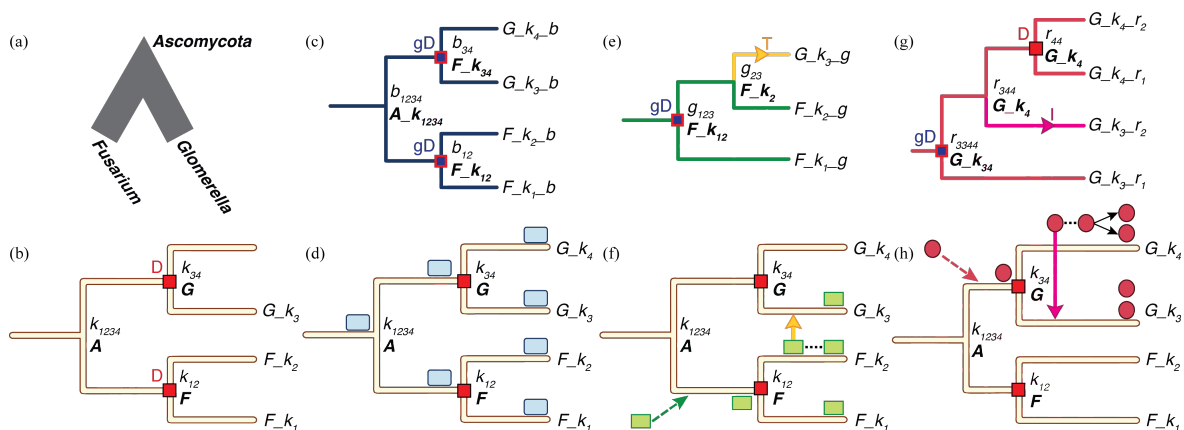


Figure 4.1: Reconstructing the composite domain shuffling history for the hypothetical family in Example 1 (Fig. 2.2). (a) The species tree. (b) The reconciled gene tree. (c) The reconciled blue domain tree. (d) The history and ancestral states of the blue domain reconstructed on the reconciled gene tree. (e) The reconciled green domain tree. (f) The history and ancestral states of the green domain reconstructed on the reconciled gene tree. (g) The reconciled red domain tree. (h) The history and ancestral states of the red domain reconstructed on the reconciled gene tree.

Input. To integrate the inferred ancestral states from individual domain families, the user must have at hand the NOTUNG-DM visual output, with the leaf and internal node names displayed. For Example 1, this includes the gene tree reconciled with the species tree, as well as the reconciled blue, red, and green domain trees. It may be useful to begin by drawing one copy of the reconciled gene tree for each domain tree. The leaves and internal nodes of the gene tree should be annotated with the names of the associated species. The user can draw the gene tree free hand or print it from NOTUNG-2.8 (Notung-2.8 Manual Sec. 3.3). For Example 1, we would now have three identical images of the gene tree, for the blue, red, and green domains, respectively. In the following steps, we will annotate these “blank” gene trees with the domain content in each contemporary and ancestral node, as well as the domain shuffling events that occurred along each branch. In a final step, merging the information from the three copies of the gene tree yields a single composite history.

The blue domain. First, we reconstruct the number of blue domains in each ancestral gene. For this reconstruction, we require (1) the association between ancestral nodes in the domain tree and nodes in the the genes and species trees and (2) the inferred domain events and, for each event, the associated branch in gene tree. All of this information is displayed in the tree panel of the reconciled domain tree (Fig. 3.9). An abstraction of this information is shown in Fig. 4.1(c).

Initialization. First, the number of blue domains in each node in the gene tree is initialized to 0.

Root. We then start the procedure by examining the root of the blue domain tree, b_{1234} . Node b_{1234} is labeled with $A_{k_{1234}}$, indicating that this ancestral blue domain was present in ancestral gene k_{1234} in *Ascomycota*. The number of blue domains at k_{1234} is set to 1. Further, gene k_{1234} is the root of the gene tree. The mapping between the root of the domain tree and the root of the gene tree indicates that the common ancestor of all genes in the family possessed a blue domain. Since the event at b_{1234} was a co-divergence, the number of blue domains in the children of k_{1234} is incremented, resulting in one blue domain in each of k_{12} and k_{34} .

Pre-order tree traversal. Next, we consider the children of b_{1234} . Node b_{12} is mapped to gene tree node k_{12} . Since b_{12} is assigned a co-divergence event, we increment the number of blue domains at the children of k_{12} , F_{k_1} and F_{k_2} . The blue domain counts at F_{k_1} and F_{k_2} are now set to 1. The depth-first traversal next visits the children of b_{12} . Since F_{k_1-b} and F_{k_2-b} are leaves, and are not associated with any events, no action is required.

The depth-first traversal then continues to node b_{34} , the other child of b_{1234} . Node b_{34} is mapped to gene k_{34} . The event at b_{34} is again a co-divergence event, and the number of blue domains at the children of k_{34} is incremented by 1. As a result, the number of blue domains at G_{k_3} and G_{k_4} is set to 1. Finally, the traversal visits the children of b_{34} . Since G_{k_3-b} and G_{k_4-b} are leaves, and are not associated with any events, the procedure terminates.

The reconstruction procedure has inferred a single blue domain in each ancestral gene in the gene tree, as shown in Fig. 4.1(d). Since only co-divergence events were inferred in the blue domain tree, no domain shuffling events are displayed on the gene tree.

The green domain. Next, we reconstruct the number of green domains in each ancestral gene. The inferred domain events and the associations between ancestral nodes in the domain tree and nodes in the gene and species trees are displayed in the tree panel of the reconciled domain tree (Fig. 3.12). An abstraction of this information is shown in Fig. 4.1(e).

Initialization. In the initialization step, the green domain counter on each node in the gene tree is set to 0.

Root. The procedure then starts at the root of the domain tree, g_{123} . The gene associated with g_{123} is k_{12} in *Fusarium*, which is not the root of the gene tree. Therefore, the number of green domains at k_{12} is set to 1, and the edge leading to k_{12} is annotated with an origination event. The event at g_{123} is a co-divergence, so the number of green domains at each of the children of k_{12} is increased from 0 to 1.

Pre-order tree traversal. The depth-first traversal proceeds to the children of g_{123} . One child, F_{k_1-g} , is a leaf in the domain tree and is not associated with an event, so no further action is taken. The other child, g_{23} , is the donor of a domain transfer event. Inspection of Fig. 3.12 reveals that G_{k_3-g} was the recipient of the transfer. The genes associated with g_{23} and G_{k_3-g} are F_{k_2} and G_{k_3} , respectively. In the gene tree, a transfer arrow is drawn from F_{k_2} , the donor, to G_{k_3} , the recipient. We then increment the number of green domains at the recipient G_{k_3} by 1.

In the final step of the depth first traversal, the procedure visits the children of g_{23} . Since both children are leaves in the domain tree and are not associated with any events, the procedure terminates.

The result of the reconstruction procedure, shown in Fig. 4.1(f), shows that a single green domain originated in ancestral gene k_{12} . This green domain was vertically inherited in F_{k_1} and F_{k_2} . The copy in F_{k_2} was then transferred to G_{k_3} , resulting in one copy of the green domain in that gene.

The red domain. Finally, we reconstruct the events and number of red domains in each ancestral gene. The inferred events and ancestral associations are displayed in the tree panel of the reconciled domain tree (Fig. 3.10). An abstraction of this information is also shown in Fig. 4.1(g).

Initialization. First, the number of red domains in each gene of the gene tree is set to 0 in the initialization step.

Root. The procedure then starts at r_{3344} , the root of the red domain tree. Node r_{3344} is labeled with gene k_{34} in *Glomerella*, so the number of red domains at k_{34} is set to 1. Since k_{34} is not the root of the gene tree, there must have been an origination event, and the edge leading to k_{34} is annotated appropriately. Because the event at r_{3344} is a co-divergence, the number of red domains in both children of k_{34} is incremented from 0 to 1.

Pre-order tree traversal. Next, the depth-first traversal considers the children of r_{3344} . Child node $G_{k_3-r_1}$ is a leaf that is not associated with any events; no further action is taken. Child r_{344} is an internal node and is the donor of an insertion event. The purple edge in Fig. 3.10 shows that $G_{k_3-r_2}$ was the insertion recipient. Domain nodes r_{344} and $G_{k_3-r_2}$ map to G_{k_4} and G_{k_3} , respectively; G_{k_4} is the gene donor of the insertion, while G_{k_3} is the gene recipient. Therefore, an arrow is added to the gene tree from donor G_{k_4} to recipient G_{k_3} . The number of red domains at recipient G_{k_3} is increased by 1, resulting in a total of 2.

The depth-first traversal then visits the children of r_{344} . Since $G.k_{3-r_2}$ is a leaf that is not associated with any events, no action is required. Internal node r_{44} , the other child, is mapped to gene $G.k_4$ and is a domain duplication node. Therefore, the number of red domains in $G.k_4$ is increased from 1 to 2, and the edge leading to $G.k_4$ is annotated with a duplication event.

Finally, the traversal visits the children of r_{44} . Since $G.k_{4-r_1}$ and $G.k_{4-r_2}$ are leaves not associated with any events, the procedure terminates.

The reconstruction procedure infers that the ancestral gene k_{34} had a single red domain, even though both of its children ($G.k_3$ and $G.k_4$) each had two copies of the red domain, as shown in Fig. 4.1(h). The copy of the red domain in k_{34} was vertically inherited by both $G.k_3$ and $G.k_4$. Then, the copy in $G.k_4$ was inserted into $G.k_3$, resulting in two different red domains in $G.k_3$. Of the two red domains in $G.k_3$, one was ancestrally inherited; the other was horizontally inherited. After the insertion, the copy in $G.k_4$ was duplicated. Thus, in contrast to $G.k_3$, the two red domains in $G.k_4$ were both vertically inherited.

Algorithm 4.1 Domain Shuffling Reconstruction

Input: $T_{GS}^* = (V_G^*, E_G^*)$; $\text{setDT} = \{T_{D_i, G}^*\} = \{(V_{D_i}^*, E_{D_i}^*)\}$

Output: The composite history of domain shuffling on the reconciled gene tree.

```

annotateGene( $T_{GS}^*$ ,  $\text{setDT}$ ) {
1   for each  $T_{D_i, G}^* \in \text{setDT}$  {
2     for each  $g \in T_{GS}^*$  {
3       // Initialize the number of instances of domain  $D_i$  in gene  $g$  to 0.
4       numDomain[ $D_i, g$ ] = 0
5     }
6      $\rho = \text{root}(T_{D_i, G}^*)$ 
7      $g = M^{DG}(\rho)$  // The gene to which the domain root is mapped.
8     numDomain[ $D_i, g$ ] = 1 // The first appearance of the domain is set to 1.
9     if  $\rho \neq \text{root}(T_{GS}^*)$  { // There was an origination event leading to  $g$ .
10      add  $\mathcal{O}$  to list events[ $D_i, g$ ]
11    } // Else, the domain was present at the family origin.
12    annotateGeneDomainNode( $\rho, g$ ) {
13  }
14 }
}

annotateGeneDomainNode( $d, g$ ) {
14 // Annotate the event on the branch leading to  $g$ .
15 add  $\mathcal{E}(d)$  to list events[ $D_i, g$ ]
16 if  $\mathcal{E}(d) = \mathcal{D}$  { // There was a domain duplication.
17   // Increment the number of domains.
18   numDomain[ $D_i, g$ ] ++
19 } else if  $\mathcal{E}(d) \in \{\mathcal{C}_S, \mathcal{C}_D, \mathcal{C}_T\}$  { // There was a co-divergence with  $g$ 
20   //  $D_i$  is passed down to both children of  $g$ .
21   // Increment the number in both children.
22   numDomain[ $D_i, r(g)$ ] ++
23   numDomain[ $D_i, l(g)$ ] ++
24 } else if  $\mathcal{E}(d) \in \{\mathcal{T}, \mathcal{I}\}$  { // There was a horizontal event from  $g$ .
25    $d^* = \text{recipient}(d)$  // Get the child of  $d$  that was the recipient.
26    $g^* = M^{DG}(d^*)$  // Consider the recipient in the gene tree.
27   // Add an edge from donor  $g$  to recipient  $g^*$  in the reconciled gene tree.
28   add  $e = (g, g^*)$  to  $E_G^*$ 
29   // Increment the number of domains in the recipient.
30   numDomain[ $D_i, g^*$ ] ++
31 } else if  $\mathcal{E}(d) = \mathcal{L}$  { // There was a loss.
32   // Decrement the number of domains.
33   numDomain[ $D_i, g$ ] --
34 }
35 if !isLeaf( $d$ ) {
36   // Traverse the tree in pre-order
37   annotateGeneDomainNode( $r(d), M^{DG}(r(d))$ )
38   annotateGeneDomainNode( $l(d), M^{DG}(l(d))$ )
39 }
}

```

Bibliography

- [1] M. Basu, E. Poliakov, and I. Rogozin. Domain mobility in proteins: functional and evolutionary implications. *Brief Bioinform*, 10(3):205–216, Jan 2009.
- [2] I. Ben-Shlomo, S. Yu Hsu, R. Rauch, H. Kowalski, and A. Hsueh. Signaling receptome: a genomic and evolutionary perspective of plasma membrane receptors involved in signal transduction. *Sci STKE*, 2003(187):RE9, Jun 2003.
- [3] M. Buljan, A. Frankish, and A. Bateman. Quantifying the mechanisms of domain gain in animal proteins. *Genome Biol*, 11(7):R74, Jul 2010.
- [4] R. Finn, A. Bateman, J. Clements, P. Coggill, R. Eberhardt, S. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. Sonnhammer, J. Tate, and M. Punta. Pfam: the protein families database. *Nucleic Acids Res*, 42:D222–D230, Jan 2014.
- [5] W. Fitch. Homology: a personal view on some of the problems. *Trends Genet*, 16(5):227–231, May 2000.
- [6] J. Gough and C. Chothia. Superfamily: Hmms representing all proteins of known structure. scop sequence searches, alignments and genome assignments. *Nucleic Acids Res*, 30(1):268–272, Jan 2002.
- [7] M. Hallett, J. Lagergren, and A. Tofigh. Simultaneous identification of duplications and lateral transfers. In *RECOMB 2004: Proceedings of the Eighth International Conference on Research in Computational Biology*, ACM Press, pages 347–356, New York, NY, USA, 2004.
- [8] C. Jones, A. Custer, and D. Begun. Origin and evolution of a chimeric fusion gene in *Drosophila subobscura*, *D. madeirensis* and *D. guanche*. *Genetics*, 170(1):207–219, May 2005.
- [9] M. Long, E. Betran, K. Thornton, and W. Wang. The origin of new genes: glimpses from the young and old. *Nat Rev Genet*, 4(11):865–75, Nov 2003.
- [10] A. Marchler-Bauer, J. Anderson, F. Chitsaz, M. Derbyshire, C. DeWeese-Scott, et al. CDD: specific functional annotation with the Conserved Domain Database. *Nucleic Acids Res*, 37:D205–D210, Jan 2009.

- [11] E. Marcotte, M. Pellegrini, M. Thompson, T. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402(6757):83–86, Nov 1999.
- [12] T. Miyata and H. Suga. Divergence pattern of animal gene families and relationship with the Cambrian explosion. *Bioessays*, 23(11):1018–27, Nov 2001.
- [13] A. Moore, A. Björklund, D. Ekman, E. Bornberg-Bauer, and A. Elofsson. Arrangements in the modular evolution of proteins. *Trends Biochem Sci*, 33(9):444–451, Sep 2008.
- [14] L. Patthy. Intron-dependent evolution: preferred types of exons and introns. *FEBS Lett*, 214(1):1–7, Apr 1987.
- [15] T. Pawson and P. Nash. Assembly of cell regulatory systems through protein interaction domains. *Science*, 300(5618):445–452, Apr 2003.
- [16] D. Sayah, E. Sokolskaja, L. Berthoux, and J. Luban. Cyclophilin A retrotransposition into TRIM5 explains owl monkey resistance to HIV-1. *Nature*, 430(6999):569–573, Jul 2004.
- [17] N. Song, J. Joseph, G. Davis, and D. Durand. Sequence similarity network reveals common ancestry of multidomain proteins. *PLoS Comput Biol*, 4:e1000063, Apr 2008.
- [18] M. Stolzer, H. Lai, M. Xu, D. Sathaye, V. B, and D. Durand. Inferring duplications, losses, transfers, and incomplete lineage sorting with non-binary species trees. *Bioinformatics*, 28:i409–i415, 2012.
- [19] M Stolzer, KM Siewert, H Lai, M Xu, and D Durand. Event inference in multidomain families with phylogenetic reconciliation. *BMC Bioinformatics*, 16:S8, 2015.
- [20] N. Vinckenbosch, I. Dupanloup, and H. Kaessmann. Evolutionary fate of retroposed gene copies in the human genome. *PNAS*, 103(9):3220–3225, Feb 2006.
- [21] C. Zmasek and S. Eddy. ATV: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, 17(4):383–4, Apr 2001.
- [22] C. Zmasek and S. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–8, Sep 2001.