The Plexus Model for the Inference of Ancestral Multi-Domain Proteins

John Wiedenhoeft, Roland Krause, Oliver Eulenstein

Abstract—Interactions of protein domains control essential cellular processes. Thus, inferring the evolutionary histories of multi-domain proteins in the context of their families can provide rewarding insights into protein function. However, methods to infer these histories are challenged by the complexity of macro-evolutionary events. Here we address this challenge by describing an algorithm that computes a novel network-like structure, called plexus, which represents the evolution of domains and their combinations. Finally, we demonstrate the performance of this algorithm with empirical data sets.

Index Terms—proteins, domains, plexus, graphs, phylogeny.

1 Introduction

Proteins containing mobile domains pose major problems to phylogenetic inference of protein families [1]. The sequencing of higher eukaryote genomes revealed that many protein coding genes bear hallmarks of hybridization and recombination events [2]. Many such evolutionary cassettes fold into sizable, distinct units that can be easily recognized in the three-dimensional structures [30]. Other recognizable and frequent elements are short motifs. Two proteins might be non-homologous for the major part of their sequence and only joined by a short mobile domain. If more domains and proteins are analyzed, it is easy to envisage a scenario where two proteins without any homology would be considered similar to each other via a third.

For phylogenetic analysis of species proteins with such promiscous domains could be removed from the data, which stabilizes the results at the expense of information content. For the task of determining evolutionary histories of families containing *multi-domain proteins* (MDPs) such information is essential, and recombination cannot simply be ignored as the vast majority of proteins in the higher eukaryotes consist of several domains [4]. As evolutionary events are rare, most of the resulting architectures are simple. However, in human about 10% of all proteins have highly complex domain architectures composed of about 200 domain families that combine frequently [5]. These MDPs are involved in essential cellular processes, including chromatin remodeling and signal transduction. Domain structure is a dominant feature for elucidation of protein-protein interaction [28].

Here, we introduce a novel graph-theoretical model, called plexus, to infer an evolutionary scenario for MDPs. A plexus is a graph that embeds the given trees by invoking macro-evolutionary events. We formulate the *MDP evolution problem* as finding the plexus with the lowest score, describe an effective heuristic to solve it and show that its implementation performs well in practice.

1.1 Related Work

After the discovery of mobile domain combinations in the 1980s, it required complete eukaryotic genome sequences for thorough investigations of the phenomenon [2]. Genome wide studies typically ignore relationships of the sequence fragments and do not attempt to map individual macro-evolutionary events.

Quantitative studies found the number of observed neighbors for a domain to follow a power-law distribution [6].

Phylogeny-oriented work concentrated on analyzing evolutionary events that establish multiple-domain compositions, and derive phylogenetic trees from domain combinations using parsimony-based criteria or clustering approaches [9], [10], [23]. Przytycka et al. used a parsimony-based approach and simplified gene fusion, domain shuffling and retrotransposition events into tractable merge and deletion operations [11]. Fong et al. constructed a more elaborate model with 3 subclasses of fusion events for MDPs to reconstruct domain trees [12]. A recent study of mechanistic events found that fusion of two genes rather than retrotransposition-based mechanisms gives rise to multi-domain proteins [25].

Previous work mostly investigated general principles of protein evolution. In contrast, methods for the reconstruction of MDP histories based on macro-evolutionary events are still in their infancy, and studies of particular protein families typically resorted to manual annotation [13], [14].

[15] suggested an approach incorporating domain histories to reconstruct ancestral domain compositions from a given collection of domain trees and a

- *J. Wiedenhoeft and R. Krause are with the Max Planck Institute for Molecular Genetics, Berlin and the Free University of Berlin. O. Eulenstein is with Iowa State University.*

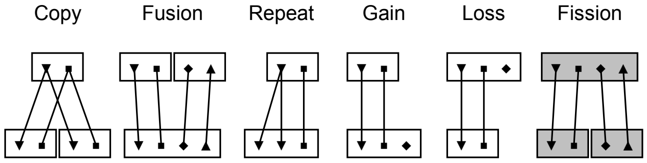


Fig. 1. Basic events in MDP evolution. Multidomain-proteins are represented by boxes containing domain nodes. Different domain families are depicted by different domain node shapes. Fission is not part of our approach, as it is modeled with elementary events (see Fig. 2), and thus depicted in grey.

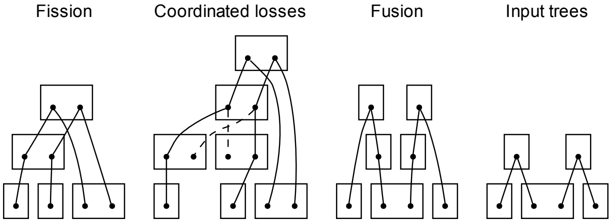


Fig. 2. Three equally plausible explanations for fission events for two domains found in three proteins, which will all yield the same phylogenetic trees and extant domain compositions (right). Note that in coordinated losses the dashed edges are not reconstructable from extant domains.

given species tree. Each domain node of a domain tree is mapped to a node of the species tree. The domains in a species node are partitioned to represent multi-domain proteins in the parent species with the weighted minimum number of merges and deletions in comparison to the child species. Their method relies on the following critical assumptions: the correctness of the domain trees, the correctness of a species tree, and the correct mapping of each domain tree node into the species tree, all of which may not be satisfiable in practice.

Suitably restricted networks to model macro-evolution events have been explored where trees are no longer sufficient and several approaches were used with success for phylogenetic displays and mapping of events, reviewed in [16]. Our approach relates to [17], which is aimed at the reconstruction of phylogenies with recombination events. However, this and similar models are not directly applicable to reconstruct the evolution of MDPs.

1.2 Our Contribution

Our novel graph-theoretic network, called *plexus* allows to describe scenarios for the evolution of a collection of domain trees. A plexus is obtained by embedding them in a gene network, which can be embedded in a species tree if one is available. Initially, only domain compositions of extant genes are known. Architectural details such as the order of the domains in a protein are not used and do not need to be specified. The initial configuration of the plexus assumes all internal genes to consist of only one domain, and extant compositions are thus modeled by multiple fusions at the leaf level, which is a highly implausible scenario. To create larger ancestral MDPs, internal nodes of the domain trees are merged in a bottom-up fashion according to a parsimony criterion.

The plexus that minimizes the overall change in MDP composition is considered to be the most plausible. As the search space of plexus topologies given a collection of domains is very large, tackling the problem requires the use of an effective heuristic, which describes the conditions under which two internal nodes in the domain trees can be merged to create an ancestral multidomain protein (d-compatibility, t-reconcilability) and a global score that needs to be minimized. The configuration obtained by gene merging reveals inconsistencies in the input trees, which are then corrected (tree correction). A final step called path detachment is used to further improve the quality of the reconstruction, since the previous steps could have lead to early fusions which introduce unnecessary domain losses.

The implementation performs well in practice, which we demonstrate on a selection of proteins with frequently recombining domains.

2 A Model for the Evolution of MDPs

The composition of extant proteins and the phylogenetic relationships between domains yield the information required for the reconstruction. In our model, evolutionary events are mapped to the domain trees and ancestral proteins are postulated. We now give an overview of the types of events and derive the plexus model.

2.1 Evolutionary Events

In the context of this work we disregard the order of domains in the proteins and consider five macroevolutionary events (Fig. 1). *Copy* events represent either gene duplication or speciation, in cases when no species tree is being used those two events go undistinguished. *Fusion* describes the union of two genes via loss of terminal and initial segments or translocation leading to an MDP. This most important event with respect to the variation in the domain composition typically involves joining adjacent genes [25]. Insertion of additional domains has been observed but contributes little variation overall [25]. *Losses* originate from truncations due to premature stop codons or silencing of exons. Many perceived losses might be missing annotations as domain prediction has a high false-negative rate [18]. The rare event of the birth of a domain is called *gain* and the introduction is mapped as the root of a domain tree. A *repeat* describes the addition of a domain (e. g. by tandem duplication). *Fission* of an MDP is a complex process requiring the gain of both a start and a stop site in the right order. The process has been hypothesized to involve reading frame shifts [19]. An alternative scenario for fission involves gene duplication with subsequent coordinated domain losses, which has been shown to explain the evolution of the monkey-king gene family [20]. A third variant explains the observations by a fusion process (see Fig. 2). We model fission by a combination of other basic events and the score for plexus is invariant to the explicit series of events.

2.2 The Plexus

Evolution of multi-domain proteins cannot be represented in a phylogenetic tree when fusions or other hybridization events need to be mapped. Phylogenetic networks have been used to infer a consensus for species evolution via gene trees or map duplication and reticulation events to a species tree [16]. We opted to model these promiscuous substructures of genes in a more refined representation, which we call *plexus*.

Generally, a plexus is a meshwork of branching and rejoining strands, e. g. a network of blood vessels in the choroid plexus or nerves in the solar plexus. The strands are not simply convoluted but have a direction as in blood flow or action potential propagation. We use this image to describe the aggregation and segregation of phylogenetic domain trees over time, forming multi-domain proteins at every convergence.

Here, a plexus is a nested graph structure, consisting of domain trees embedded into a gene network and an optional species tree. To construct it, inner nodes in the domain trees are joined into gene nodes to create ancestral domain compositions such that their connections can be described with one of the evolutionary events above. Additional internal nodes on the edges can be introduced where necessary.

The true sequence of evolutionary events is unknown. It might contain sequences of events such as birth and complete disappearance of a domain family that cannot be recovered from the extant sequence data and domain compositions. This hypothetical construct is called *expanded plexus* and describes the putative true sequence of MDP evolution events (see Fig. 3, left side and Fig. 1).

We call a plexus that is derived from a richer model *counterpart*. The *reconstructable counterpart* of an expanded plexus is constrained to MDP evolution events that we could infer in principle. However, the repeated loss and gain of domains could lead to arbitrarily long paths. We can restrict the topologies to a finite number by requiring that each gene node must contain at least one domain node with out-degree ≥ 2, which makes the number of domain tree nodes an upper bound to the number of gene nodes. By contracting out-edges of gene nodes containing only domain nodes of out-degree 1, we transform a reconstructable plexus into its *compact counterpart*. In Fig. 3 (middle), these are the gene edges made up by the dashed edges.

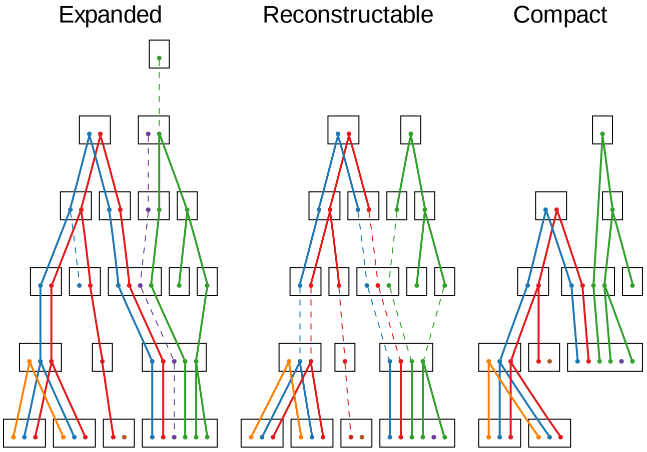


Fig. 3. Counterparts of plexūs for a set of MDPs and their domain trees. Edges mark inheritance relations between domains (dots) in genes (rectangles). Different domain trees are shown in different colors. The expanded plexus consists of evolutionary events (see Fig. 1). Its non-reconstructable edges are dashed and disappear in the reconstructable counterpart. Some of the resulting inner gene nodes may then contain only domain nodes with combined in- and out-degree ≤ 1. Contracting their out-edges (dashed) results in the compact counterpart.

Reconstruction of a compact plexus is therefore the aim of this endeavour and the problem is reduced to partitioning domain tree nodes. It is still infeasible to evaluate all potential partitions and we have developed a heuristic to find the best scoring topology. In the following section, we will give a more rigid formalization in order to derive the scoring scheme and the heuristic.

3 Reconstruction of the Compact Plexus

3.1 Basic Definitions and Notation

Let $G := (V, E)$ be a directed acyclic graph (DAG) with an edge set $E(G)$ and a node set $V(G)$. Note that rooted phylogenetic trees and networks are DAGs. We denote the in-degree and the out-degree of a node $v \in G$ by $\deg^-(v)$ and $\deg^+(v)$ respectively. The *edge contraction* of an edge $(v, w) \in E$ is achieved by first identifying v with w, and then deleting the resulting loop. For nodes $v, w \in V$ and $j \in \mathbb{Z}^+ \cup \{\infty\}$ we (i) write $v \sim_j w$, if $v \neq w$ and there is a path from v to w of at most j edges in G, and (ii) define $v \sim_{-j} w := w \sim_j v$. If $v \sim_k w$ and $k > 1$, we write $v > w$ and call v a *predecessor* of w, and w a *successor* of v. In the case $k = 1$, we use the terms *direct predecessor* or *parent* and *direct successor* or *child* accordingly. When using the plural predecessors/successors, direct predecessors/successors are included. We say that v and w are *connected* if $v \sim_{\pm\infty} w$. We call $\mathrm{LCA}\{v_1, \ldots, v_k\} := \min\{v \in V(G) \mid v_i < v, 1 \leq i \leq k\}$ the *least common ancestor (LCA)* of $\{v_1, \ldots, v_k\}$. Let $k, l \in \mathbb{Z} \cup \{-\infty, \infty\}$, then we define the *k-neighborhood* of a set $U \subseteq V$ to be $\mathrm{N}^k(U) := \{v \in V \mid \exists u \in U : v \sim_k u\}$, and $\mathrm{N}^{l,k}(U) := \mathrm{N}^k(\mathrm{N}^l(U))$. For instance, given a directed path $a \to b \to c$, $\mathrm{N}^2\{a\} = \{b, c\}$ and $\mathrm{N}^{1,1}\{a\} = \mathrm{N}^1\{b\} = \{c\}$.

3.2 Plexus and Evolutionary Events

We first formalize the plexus as a model for MDP evolution.

Definition 1 (Plexus):

- Let D be a directed forest of phylogenetic domain trees such that all nodes of a tree belong to the same domain family and no two domains of the same family belong to different trees. Let a DAG G represent a phylogenetic network of genes, and S a species tree. d denotes domain nodes, g gene nodes and s species nodes.
- Let $\gamma\colon V(D) \to V(G)$ be a function that maps domains to genes, and $\Delta(g) := \{d \in V(D) \mid \gamma(d) = g\}$ be the set of all domains in gene $g \in V(G)$.
- Similarly, let $\sigma\colon V(G) \to V(S)$ be the species of a gene, and $\Gamma(s) := \{g \in V(G) \mid \sigma(g) = s\}$ all genes in species s.

We call a tuple $(D, G, S, \gamma, \sigma)$ a *plexus* iff

- There are no empty genes: $|\Delta(g)| \neq 0$ for all $g \in V(G)$.
- There is a gene edge if and only if there is at least one domain edge between the domains in the two genes: $\exists (g_1, g_2) \in E(G) \Leftrightarrow \exists (d_1, d_2) \in E(D) : \gamma(d_1) = g_1, \gamma(d_2) = g_2$.
- Each domain maps to a gene higher than those of its successors: $\forall d_1, d_2 \in V(D) : d_1 > d_2 \Rightarrow \gamma(d_1) > \gamma(d_2)$.
- Each gene maps to the least common ancestor species of all its successor's species or higher: $\forall g \in V(G) : \sigma(g) \geq \mathrm{LCA}\{\sigma(g_1), \ldots, \sigma(g_n)\}$ for all successors $g_1, \ldots, g_n$ of gene g.

As a preliminary for defining evolutionary events, let $\nu^+(g_1, g_2) := \{p \in \Delta(g_1) \mid \exists c \in \Delta(g_2) : (p, c) \in E(D)\}$ be the set of domains in gene g_1 that have children in gene g_2, and $\nu^-(g_1, g_2) := \{c \in \Delta(g_1) \mid \exists p \in \Delta(g_2) : (p, c) \in E(D)\}$ the set of domains in gene g_1 with parents in g_2. We then provide a formalization of the events shown in Fig. 1 except fission, which is not used in our model.

Definition 2 (MDP evolution events):

- Let $(g_1, g_2) \in E(G)$ and $\deg^+(g_1) = \deg^-(g_2) = 1$. Let $k \in \mathbb{N}$, We call $\{g_1, g_2\}$ a *loss event of size k* if $|\Delta(g_1)| - k = |\Delta(g_2)| = |\nu^+(g_1, g_2)| = |\nu^-(g_2, g_1)|$.
- Similarily, we call $\{g_1, g_2\}$ a *gain of size k* if $|\Delta(g_1)| = |\Delta(g_2)| - k = |\nu^+(g_1, g_2)| = |\nu^-(g_2, g_1)|$.
- A gain or loss of size 0 is called a *neutral event*.
- We call $\{g_1, g_2\}$ a *repeat of size k* and g_1 a *repeat node* if $\Delta(g_1) = \nu^+(g_1, g_2)$, $\Delta(g_2) = \nu^-(g_2, g_1)$ and $|\Delta(g_1)| + k = |\Delta\, g_2|$.
- Let $C := \{(g_1, g_i)\}_{i=2}^k \subseteq E(G)$, $\deg^+(g_1) = k - 1$ and $\deg^-(g_2) = \ldots = \deg^-(g_k) = 1$. We call C a *copy event* if $|\Delta(g_1)| = \ldots = |\Delta(g_k)|$ and all domains in $\Delta(g_1)$ have exactly one direct successor in each $\Delta(g_2), \ldots, \Delta(g_k)$.
- Let $F := \{(g_i, g_1)\}_{i=2}^k \subseteq E(G)$, $\deg^-(g_1) = k-1 > 1$ and $\deg^+(g_2) = \ldots = \deg^+(g_k) = 1$. We call F a *fusion event* if $\bigcup_{i=2}^k \nu^-(g_1, g_i) = \Delta(g_1)$ and all domains in $\Delta(g_2), \ldots, \Delta(g_k)$ have exactly one direct successor in $\Delta(g_1)$.

Loss, gains, fusions, copies and repeats are refered to as MDP evolution events.

Let P be a plexus. We call P *expanded* if for each of its gene nodes $g_1 \in V(G)$ either $\{(g_1, g_2) \mid g_2 \in \mathrm{N}^1\{g\}\}$ or $\{(g_2, g_1) \mid g_2 \in \mathrm{N}^{-1}\{g_1\}\}$ is an MDP evolution event. P is called *reconstructable* if no non-terminal gene node contains any domain node for which the sum of its in- and out-degree is ≤ 1. A reconstructable plexus P_R is called the *reconstructable counterpart* of an expanded plexus P iff it can be obtained by subsequently deleting any non-terminal nodes that have a combined in- and out-degree ≤ 1 and their incident edges. Let $(g_1, g_2) \in E(G)$ such that $\forall d_1 \in \nu^+(g_1, g_2) : \deg^+(d_1) = 1$ and $\forall d_2 \in \nu^-(g_2, g_1) : \deg^-(d_2) = 1$. (g_1, g_2) is called *contractible*. The operation of contracting all $(d_1, d_2) : \gamma(d_1) = g_1, \gamma(d_2) = g_2$ and merging g_1 with g_2 is called *gene edge contraction*. A plexus P_C is said to be *contracted* if it contains no contractible gene edges, and *contracted counterpart* of a plexus P if it is contracted and can be obtained by contracting gene edges in P. This is similar to the concept of *minors* in undirected graphs. A contracted plexus P_C is called the *compact counterpart* of a plexus P, if there is a reconstructable counterpart P_R of P such that P_C is a contracted counterpart of P_R.

3.3 Scoring Evolutionary Scenarios

To measure the quality of our reconstruction, we introduce a score on the compact plexus that considers evolutionary events by a unified criterion. Only losses, gains and fusions are events in which gene nodes connected by a gene edge contain nodes that are not related to any node in the other gene node. In contrast to copy and repeat, the direct successor gene nodes are intrinsically different from their predecessors. The number of these domains is therefore a good measure to model evolutionary changes.

Unfortunately, compactification imposes contraction to gene edges in fusion, gain and loss, and hence to exactly those events that we consider to be of evolutionary importance. However, we can reconstruct these events from the compact counterpart.

The number of losses accounting for an edge $(g_1, g_2) \in E(G)$ is $|\Delta(g_1)| - |\nu^+(g_1, g_2)|$. Fusions and addition of a gained domain to an existing MDP can be calculated similarly by $|\Delta(g_2)| - |\nu^-(g_2, g_1)|$ for edges $(g_1, g_2) \in E(G)$. It is noteworthy to mention that the order of fusions is lost during compactification but the number of domain changes depends on that order. For example consider a gene node with in-degree 3 and predecessor gene nodes of size 1, 2 and 3. Combining genes of sizes 1 and 2 first creates an out-edge of size 3, and then merging with the third gene node creates an out-edge of size 6. In contrast, combining genes of sizes 2 and 3 first produces out-edges of size 5 and 6, so the score would have to be 2 edges higher. In other words, there are reconstructable plexūs with different fusion sequences that have the same compact counterpart. The fusion score defined above is higher than that of any fusion order in the expanded plexus, since it equals $(\deg^-(g_2) - 1) \cdot \sum_{g_1 \in \mathrm{N}^{-1}\{g_2\}} |\nu^-(g_2, g_1)|$. Under the assumption that successive fusion events are typically rare, we tolerate this overestimate.

The compact plexus might contain gene edges which are transitive with respect to the species tree, i. e. an edge (g_1, g_2) for which $\sigma(g_2) \notin \{\sigma(g_1)\} \cup \mathrm{N}^1\{\sigma(g_1)\}$. The intermediate genes that initially mapped to the species between $\sigma(g_1)$ and $\sigma(g_2)$ got lost during compactification. Nevertheless, these genes need to be reflected in the score, since they were lost in all but one speciation lineage in every intermediate species. To quantify the number of lineages in which a gene got lost, we define the following:

Definition 3 (Path reductivity): Let A be a DAG. For each directed path $(v_1, \dots, v_n) \in A$ we define the *path reductivity*

$$|(v_1, \dots, v_n)| := \sum_{i=2}^{n-1} (\deg^+(v_i) - 1)$$

Then $\pi(v_1, v_n)$ is defined to be a minimal path from v_1 to v_n with respect to path reductivity. If more than one such path exists, one is chosen by chance.

Note that only nodes *between* v_1 and v_n are included, since only these intermediate nodes induce losses. We will use this definition for whole gene losses in scoring as well as in our heuristic to estimate the number of domain losses we induce during reconstruction. The number of gene losses along a gene edge due to speciation events between $\sigma(g_1)$ and $\sigma(g_2)$ is $|\pi(\sigma(g_1), \sigma(g_2))|$. To score such an event, we multiply it by the number of domains being lost. If no repeat is involved, this can be described as the number of domain edges between these two genes, which equals $|\nu^+(g_1, g_2)|$ or $|\nu^-(g_2, g_1)|$. In repeats, the latter will be larger by the number of additional domains. Since g_1 could then be mapped to the same species as g_2 due to its out-degree being 1, we use the first, hence $|\nu^+(g_1, g_2)| \cdot |\pi(\sigma(g_1), \sigma(g_2))|$.

Combining the above and summing over all edges defines a parsimony score which measures changes in domain composition over evolutionary time:

Definition 4 (Compact plexus score): Let $P = (D, G, S, \gamma, \sigma)$ be a compact plexus. Then the *compact plexus score* for P, denoted by $S(P)$, is defined as

$$\sum_{(g_1, g_2) \in E(G)} \Big[|\Delta(g_1)| + |\Delta(g_2)| - \big|\nu^-(g_2, g_1)\big| \\ + (|\pi(\sigma(g_1), \sigma(g_2))| - 1) \big|\nu^+(g_1, g_2)\big| \Big]$$

As each $|\Delta(g_i)|$ is added for each in- and out-edge, the score can be split into a node part S_V and an edge part S_E:

$$S := S_V + S_E$$

with

$$S_V := \sum_{g \in V(G)} (\deg^-(g) + \deg^+(g)) \, |\Delta(g)|$$

and

$$S_E := \sum_{(g_1, g_2) \in E(G)} \Big[(|\pi(\sigma(g_1), \sigma(g_2))| - 1) \big|\nu^+(g_1, g_2)\big| \\ - \big|\nu^-(g_2, g_1)\big| \Big]$$

In principle, events could have different probabilities, which would be hard to compute in the absence of benchmark data. Fusion and large loss events are known to be rare, which is reflected in their high score [7].

Given the scoring scheme above, we now define the following problem:

Problem 1 (MDP evolution): Instance: A forest D of domain trees, a species tree S (possibly a single null node) and a partition L of their combined leaf set such that each set gene node corresponds to a known MDP composition. *Find:* A compact plexus P in which L is the leaf gene node set and which displays D such that the plexus score $S(P)$ is minimal.

4 Our Solution

The definition of our reconstruction problem above applies to input trees free of errors. It is unknown whether there is an analytical solution within acceptable run-time complexity for undistorted input. A thorough evaluation would be worthwhile but is beyond the scope of this work. In real applications the input trees typically contain numerous wrong splits. Trees built on domains use less information than trees on full-length proteins simply because they are shorter. As this is a problem which is hard to avoid, principles for building a heuristic will have to be derived from targeting it specifically. As a consequence,

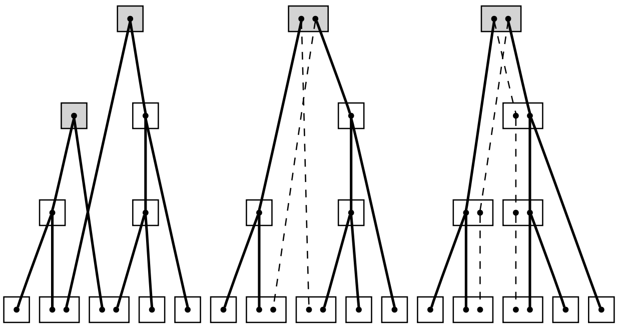


Fig. 4. Gene node merging and transitive gene edge reduction. On the left, the two grey gene nodes are separated. In the middle, they have been merged into a single gene, now containing all domains of the two original genes. The merged node's out-degree is now 4. Reducing gene edges by piping domain edges through gene nodes along the transitive path (right side), it is reduced to 2 again.

these heuristics might not be optimal for undistorted input, but perform well in practice.

Our method employs a greedy approach, aimed specifically at dealing with real data. In order to minimize the costly fusion events, we merge gene nodes according to a compatibility criterion, which is defined by whether a reduction of transitive edges below the merged gene results in an out-degree not greater than the maximum out-degree of the original gene nodes (d-compatibility). This transitive reduction is accompanied by a change in the domain-to-gene mapping γ, which introduces domain losses along transitive paths. The number of such domain losses defines the cost for merging two genes, which serves as a key in a priority queue containing gene pairs. A second criterion aims at minimizing the introduction of compositions that would lead to a global increase in domain losses (t-reconcilability). After each merge, and mapping of the new gene to the LCA species of its successors, a number of keys have to be updated, since new compatibilities and different costs (usually smaller) could have been introduced. We show that the number of required updates is rather small. The heuristic converges quickly to a state in which only incompatible gene node pairs remain in the queue, which is the termination criterion.

We now provide a detailed description of the concepts being employed in our heuristic.

4.1 Gene Node Merging

Definition 5 (Gene node merging): Let $g_1, g_2 \in V(G)$. Then the *gene node merging* $\sqcup\colon V(G) \times V(G) \to V(G)$ is defined by adding edges (g_{12}, c) for all $c \in \mathrm{N}^1\{g_1\} \cup \mathrm{N}^1\{g_2\}$ and (p, g_{12}) for all for all $p \in \mathrm{N}^{-1}\{g_1\} \cup \mathrm{N}^{-1}\{g_2\}$ to $E(G)$, changing the mapping γ such that $\gamma(g_1)$ and $\gamma(g_2)$ becomes $\gamma(g_{12})$, and $\sigma(g_{12}) := \mathrm{LCA}\{\sigma(g_1), \sigma(g_2)\}$, and finally deleting g_1 and g_2 from $V(G)$.

The process is illustrated in Fig. 4 (left and middle). As can be seen, the out-degree of the newly created gene can exceed the out-degree of the genes it replaces. Not only does this introduce poorly resolved copy events, it also induces new fusions. This happens if the merging induces *transitive edges*:

Definition 6 (Transitive edge and path): Let G be a graph. An edge $(v_1, v_2) \in E(G)$ is called *transitive* if $v_2 \in \mathrm{N}^k\{v_1\}$ for any $k > 1$. A *transitive path* of a transitive gene edge (v_1, v_2) is any path $(v_1, \dots, v_2)$.

Whenever transitive edges occur, we can decrease the out-degree again by applying transitive reduction to transitive edges. Instead of just deleting gene edges – and domain edges along with them – we "redirect" the domain edges through genes along the transitive path, introducing additional domain losses (Fig. 4, right).

Definition 7 (Transitive gene edge reduction): Let $(g_1, g_k) \in E(G)$ be a gene edge, and $\pi(g_1, g_k)$ be a transitive path $(g_1, \dots, g_k)$ of minimal reductivity with length k. The transitive gene edge reduction is performed by replacing (g_1, g_k) by a path $(g_1, g'_2, \dots, g'_{k-1}, g_k)$, splitting domain edges in $\Delta(g_1) \times \Delta(g_2)$ accordingly such that $d'_i \in \Delta(g'_i)$ and merging g_i with g'_i for all $1 < i < k$.

Since every domain is lost $\deg^+(g) - 1$ times in every gene node g along a transitive path, choosing $\pi(g_p, g_c)$ to be the path into which an edge is reduced locally minimizes the number of domain losses, since it is the path of minimal reductivity:

Definition 8 (Reduction cost of a gene edge): Let (g_1, g_2) be a gene edge reduced into a minimum reductivity path $\pi(g_1, g_2)$. The *reduction cost* is defined to be the number of additional domain losses being induced during reduction, which equals

$$\begin{aligned} r(g_1, g_2) := \;& |\pi(g_1, g_2)|\, |\nu^+(g_1, g_2)| \\ = \;& |\pi(g_1, g_2)|\, |\nu^-(g_2, g_1)| \end{aligned}$$

if both gene nodes are not repeat nodes, and 0 otherwise.

In order not to merge any combination of gene nodes, we require that the out-degree which can be obtained by transitive reduction does not exceed the maximum out-degree of the two nodes. Instead of enforcing the input trees to be binary, this allows for local differences in the tree resolution (see Fig. 5).

Theorem 1 (Minimal obtainable out-degree): Let $g_1, g_2 \in E(G)$ be two irreducible gene nodes such that $g_1 \notin \mathrm{N}^\infty\{g_2\} \cup \mathrm{N}^{-\infty}\{g_2\}$. Let g_{12} be a gene node obtained by merging g_1 and g_2. Then the minimal out-degree $\deg^\triangleleft(g_{12})$ that can be obtained by a sequence of transitive reductions to out-edges of g_{12} is $\deg^\triangleleft(g_{12}) := \left|\mathrm{N}^1\{g_1\}\right| + \left|\mathrm{N}^1\{g_2\}\right| + \left|\mathrm{N}^1\{g_1\} \cap \mathrm{N}^1\{g_2\}\right| - \left|\mathrm{N}^1\{g_1\} \cap \mathrm{N}^\infty\{g_2\}\right| - \left|\mathrm{N}^\infty\{g_1\} \cap \mathrm{N}^1\{g_2\}\right|$.

Proof: $g_1 \notin \mathrm{N}^\infty\{g_2\} \cup \mathrm{N}^{-\infty}\{g_2\}$ ensures that no cycles are introduced and the gene graph remains

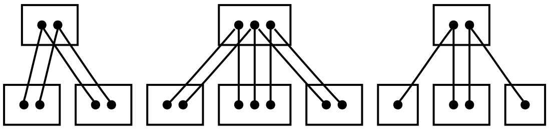


Fig. 5. In the left figure, merging of two genes has not increased the out-degree. In the center, though two trees are binary, the merge result has an out-degree of three as this does not exceed the out-degree of the third tree. On the right hand side however, an out-degree of three exceeds that of all merged trees and is thus forbidden.

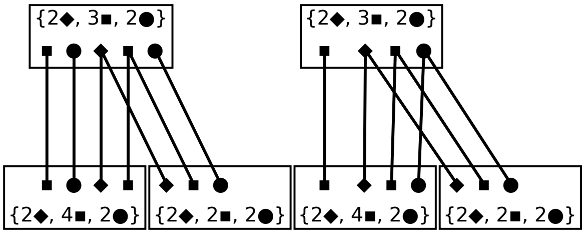


Fig. 6. Composition profiles. In the reconstruction of a gene node we show a typical artifact from errors in the phylogenetic reconstruction that leads to additional domains in the ancestral gene, here the bullet-shaped domain nodes in the left figure. Despite the left variant containing two copies of that domain family the outdegree-profile (in braces) for both variants is identical.

acyclic. $\left|\mathrm{N}^1\{g_1\}\right| + \left|\mathrm{N}^1\{g_2\}\right|$ is the out-degree of g_{12} before any transitive reduction and disregarding any multiple edges (children are treated as different even if they are the same). Any outgoing edge from g_1 points to an element $c_i \in \mathrm{N}^1\{g_1\}$ by definition. For all $c_i \in \mathrm{N}^\infty\{g_2\}$ there is a path from g_2 into which their incoming edges can be reduced. The same argument holds for $\mathrm{N}^\infty\{g_1\} \cap \mathrm{N}^1\{g_2\}$. Since $\mathrm{N}^1\{g_1\} \subseteq \mathrm{N}^\infty\{g_1\}$ and $\mathrm{N}^1\{g_2\} \subseteq \mathrm{N}^\infty\{g_2\}$, we account for that there can only be one edge between any two genes. However we subtracted $\left|\mathrm{N}^1\{g_1\} \cap \mathrm{N}^1\{g_2\}\right|$ twice as either edge is being reduced into the other, so we have to compensate by adding it one time. □

We can find all pairs of gene nodes that can be merged without violating neither out-degree nor compact plexus properties by the following criterion:

Definition 9 (d-compatibility): Two irreducible gene nodes g_1, g_2 are called *d-compatible* if $\deg^{\triangleleft}(g_1 \sqcup g_2) = \max\{\deg^+(g_1), \deg^+(g_2)\} \le d$, i. e. one can obtain a gene node with an out-degree that does not exceed neither the largest of the original nodes nor a given upper bound by merging g_1 and g_2 and applying a sequence of transitive reductions to the merged gene node. However, if either of them is a leaf, or $g_1 \notin \mathrm{N}^\infty\{g_2\} \cup \mathrm{N}^{-\infty}\{g_2\}$ (gene nodes are related), then they are incompatible. If exactly one of the gene nodes is a repeat node, it is only compatible if its direct successor is also a direct successor of the other node.

The latter condition avoids repeat nodes g_t being merged with nodes far up in the plexus, since otherwise they would be compatible to any $g \in \mathrm{N}^{1,-\infty}\{g_t\}$ and induce many losses. Interestingly, for gene node merging we have to consider exactly those evolutionary events we do not consider for our scoring scheme, and vice versa. Also, d-compatibility prevents the introduction of fission events, as all domain nodes would have on out-degree ≤ 1, but the merged gene's out-degree would be 2. Choosing some $d \neq \infty$ is usually not required and will only have some positive effect if one of the domain trees is very poorly resolved. Nodes with high out-degree would then tend to falsely attract a number of other gene nodes. Setting d to a reasonable value will take care that poorly resolved nodes are not being merged at all.

d-compatibility alone leads to domain compositions that do not resemble recent MDPs, leading to many losses as seen in Fig. 11(b). Many compatibilities arise merely by chance or by false tree splits. We therefore ensure that gene nodes *resemble recent compositions* by the following:

Definition 10 (Composition profile): Let $M = \{d_1, \dots, d_k\}$ be a set of domains in a gene node. M is partitioned into subsets $\{F_1, \dots, F_m\}$ of nodes that belong to the same input tree. The set of domain families is denoted by representatives $p := \{\mathcal{F}_1, \dots, \mathcal{F}_m\}$. Let $m(\cdot) : \mathcal{F}_i \rightarrow \mathbb{N}$ be the mapping $m(\mathcal{F}_i) = 2\left|\{n \in F_i \mid \deg^+(n) = 0\}\right| + \sum_{d \in F_i} \deg^+(d)$. Then (p, m) is called the *composition profile* of M.

Definition 11 (t-reconcilability): A profile p_1 is called *t-reconcilable* to a profile p_2 if it does not contain any domain not in p_2 and $\forall \mathcal{F}_i^1 \in p_1 : \exists \mathcal{F}_i^2 \in p_2 : \mathcal{F}_i^1 = \mathcal{F}_i^2, m(\mathcal{F}_i^1) \le m(\mathcal{F}_i^2) + t$, where t is a non-negative integer describing a chosen tolerance value.

Simply put, a value is assigned to each domain family that describes how often a domain of this family occurs in a composition. Those without children are given the same value as those with two children, those with just one child are weighted half. A gene node in a compact plexus will either contain only nodes without children, or no node without children. The reasoning behind this definition is illustrated in Fig. 6: on the right the upper gene node resembles its left direct successor, whereas on the left it contains one bullet-shaped domain more than any of its direct successors. Both predecessor gene nodes have the same profile, since in the left subfigure each bullet-shaped domain has only one out-edge. We might call this a *coordinated loss* of this domain; it can be caused by a tree root being placed in the gene node above, but will often occur due to false tree splits. These might introduce disruptions to the optimal topology. t-reconcilability aims to compensate for this, while providing a concept of similarity to recent

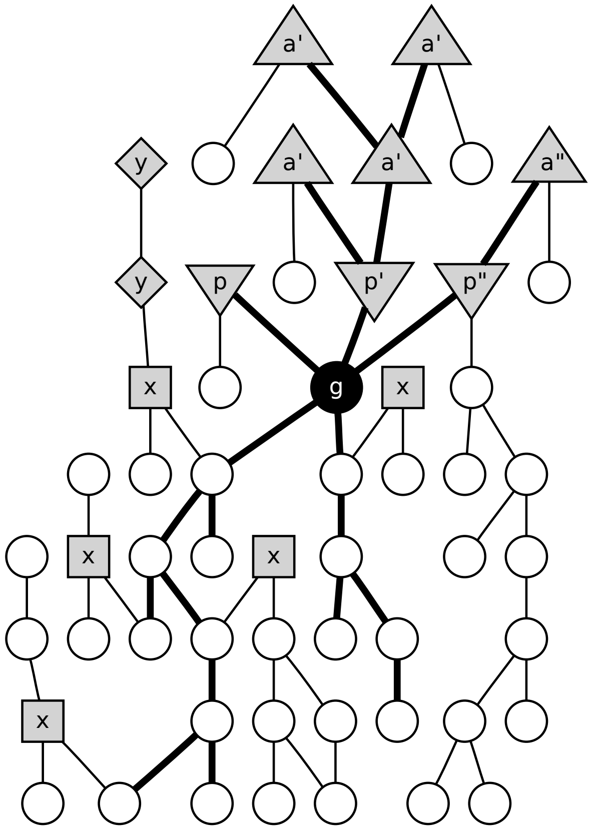


Fig. 7. A part of a gene graph G after a merging two genes, resulting in gene g. Paths through this gene are represented by bold edges. Grey nodes represent those genes for which d-t-distance has to be reevaluated in the following combinations: $\{g\} \times \{x, y\}$, $\{p\} \times \{p', p'', a', a''\}$, $\{p'\} \times \{p, p'', a''\}$, $\{p''\} \times \{p, p', a'\}$, $\{x\} \times \{p, p', p'', a', a''\}$. For the other nodes, denoted by $\circ$, there is no other node for which g could have induced any changes in their d-t-distance. In this example, we would only have to reevaluate 63 out of 1540 pairs (≈4.1%).

compositions. One should choose t to be small to avoid meaningless ancestral compositions and large loss counts, since every domain that exceeds any extant composition will have to be lost at some point. Notice that, in contrast to d-compatibility minimizing local loss counts, t-reconcilability aims on globally avoiding the introduction of foreseeable losses. On the other hand, setting $t = 0$ assumes that the topologies of all input trees are correct, which will rarely be the case. The larger t, the more the solution will tend towards one that would be obtained by d-compatibility alone and would thus again allow random compatibilities to play their role. $t = 1$ typically yields the best results. Combining d-compatibility and t-reconcilability provides us with a criterion for the merges to prefer and to avoid:

Definition 12 (d-t-distance): If two gene nodes g_1, g_2 are d-compatible and the profile of $g_1 \sqcup g_2$ is t-reconcilable to a profile of any input composition, their *d-t-distance* $c(g_1, g_2)$ is $r(g_1 \cup g_2)$, otherwise it is ∞.

d-t-distance provides a measure of preference for merging genes. We use it as key for pairs of genes in a priority queue. The actual heuristic is fairly simple: pick the pair with the shortest d-t-distance and merge it, until the shortest distance is ∞. By merging, the topology of the plexus changes, and the d-t-distance of some pairs might change, most notable for two nodes becoming compatible since the newly merged node induces a path between their direct successors. Recalculating costs for all pairs would be inefficient and priorization would no longer be efficient.

An *a-priori* set of candidates excluding all gene nodes that cannot be compatible with the current gene node ensures a tractable solution space. d-t-distance is directly tied to d-compatibility, which again is determined by the reducing paths involved. We therefore have to recalculate all pairs of genes that will be affected by newly created paths through g_{12} (see Fig. 7). Trivially, predecessors of g_{12} are incompatible to successors, and g_{12} is incompatible to both, so these updates can be made without recalculation. The pairs that have to be recalculated are the following:

Theorem 2 (Merge-affected gene node pairs): Let g be the gene node resulting from a merge. Then the set A of gene node pairs affected in terms of their d-t-distance is the union of the three sets

$$\{g\} \times [(\mathrm{N}^{\infty,-1}\{g\} \cup \mathrm{N}^{1,-\infty}\{g\}) \setminus (\mathrm{N}^{\infty}\{g\} \cup \mathrm{N}^{-\infty}\{g\})]$$

$$\mathrm{N}^{-\infty}\{g\} \times [\mathrm{N}^{\infty,-1}\{g\} \setminus \mathrm{N}^{\infty}\{g\}]$$

$$\forall g_{-1} \in \mathrm{N}^{-1}\{g\} : \{g_{-1}\} \times [\mathrm{N}^{-\infty}\{g\} \setminus \mathrm{N}^{-\infty}\{g_{-1}\}]$$

Proof: Let $g_k \in \mathrm{N}^k\{g\}$. (1) The newly created gene node g could be compatible to any $g_{\infty,-1}$, since there would be a reducing path $g, \ldots, g_\infty$ such that $(g_{\infty,-1}, g_\infty) \in E(G)$, hence $\mathrm{N}^{\infty,-1}\{g\}$ is a set of potentially d-compatible nodes. The same argument holds for any $g_{1,-\infty}$, which implies that $(g_{1,-\infty}, \ldots, g_1)$ is a reducing path to $(g, g_1) \in E(G)$. Since $\mathrm{N}^{-\infty}\{g\} \cup \mathrm{N}^{\infty}\{g\}$ is the set of all nodes connected to g, these can be excluded from the candidate set, hence $(\mathrm{N}^{\infty,-1}\{g\} \cup \mathrm{N}^{1,-\infty}\{g\}) \setminus (\mathrm{N}^{\infty}\{g\} \cup \mathrm{N}^{-\infty}\{g\})$ contains all nodes possibly d-compatible to g. (2) g might have connected paths that could possibly be reducing. Any direct predecessor of nodes on any path through g could form a reducible edge with that node. Thus any node $g_{-\infty}$ could be compatible with any $g_{\infty,-1}$, unless $g_{\infty,-1}$ is connected to $g_{-\infty}$, which is the case iff $g_{\infty,-1} \in \mathrm{N}^{\infty}\{g\}$. Thus, $\mathrm{N}^{-\infty}\{g\} \times [\mathrm{N}^{\infty,-1}\{g\} \setminus \mathrm{N}^{\infty}\{g\}$ contains all pairs for which g potentially induces reducing paths. (3) g is also a direct successor of each g_{-1}, thus $(g_{-1}, g) \in E(G)$ is an edge that is possibly reducible by a path $g_{-\infty}, \ldots, g$, unless it is a predecessor of g_{-1}, since connected nodes are d-incompatible. Hence $\mathrm{N}^{-\infty}\{g\} \setminus \mathrm{N}^{-\infty}\{g_{-1}\}$ is the set of possibly d-compatible nodes for a g_{-1}. □

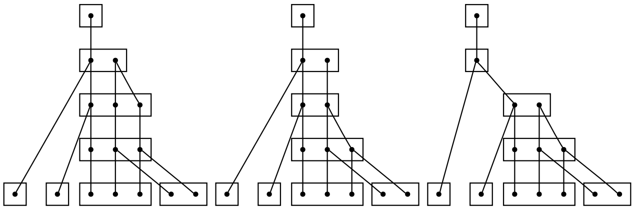


Fig. 8. Example for two steps of tree correction.

Initially, we alternate between transitive reduction of all gene nodes and merging the two gene nodes with the shortest d-t-distance, until there is no pair whose distance is $< \infty$. We avoid merging repeat gene nodes with copy gene nodes and thus violating compact plexus properties by inserting an additional gene node in the copy gene node's out-edge and merging it with the repeat gene node.

4.2 Postprocessing

Two simple procedures can lower the number of losses and curate the compact plexus.

Tree Correction: The above heuristic can introduce gene nodes below existing ones, pushing some tree nodes higher during gene node merges, thereby *stretching* subtrees which introduces coordinated losses. The same effect can be observed for false tree splits. As compensation, we introduce *tree correction* (Fig. 10): consider the coordinated loss of the bullet-shaped domains in Fig. 6. The obvious solution would be to combine the two domain nodes on the left side into one as shown on the right side. This however can only be done if they have the same parent or their respective parents can also be merged. This gives rise to a recursive definition of domain compatibility. Let d_1, d_2 be two domains of the same domain tree which are mapped to the same gene $g = \gamma(d_1) = \gamma(d_2)$. d_1 and d_2 are said to be compatible domains if (1) both have exactly one child, and the child of d_1 is in a different child gene of g than d_2 (as shown in Fig. 6), or (2) for each child gene g_c of g, assuming w.l.o.g. that d_1 has at most as many children in g_c as d_2, there is a compatible child of d_2 for each child of d_1. Leaf domains are always incompatible. More formally, let $\mathrm{N}^1_c\{d_1\}, \mathrm{N}^1_c\{d_2\}$ be the children of d_1, d_2 in g_c, and let $\left|\mathrm{N}^1_c\{d_1\}\right| \leq \left|\mathrm{N}^1_c\{d_2\}\right|$. Let B be a bipartite graph whose node sets are $\mathrm{N}^1_c\{d_1\}$ and $\mathrm{N}^1_c\{d_2\}$, and there is an edge if and only if two domains are compatible. We say that g_c is resolved with respect to d_1, d_2 if $\mathrm{N}^1_c\{d_1\}$ is completely contained in a maximal matching in B. d_1, d_2 are compatible if all children of g are resolved with respect to d_1, d_2. Tree correction starts by merging compatible domains which have the same parent, and recursively merges compatible children until no further changes can be made (Fig. 8). Subsequently, compactification is applied to ensure a compact plexus.

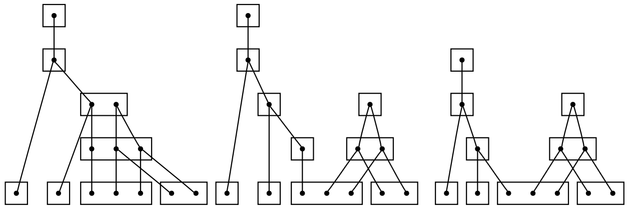


Fig. 9. Example for path detachment. Further losses are removed by a maximal split of domain, followed up by gene edge contraction.

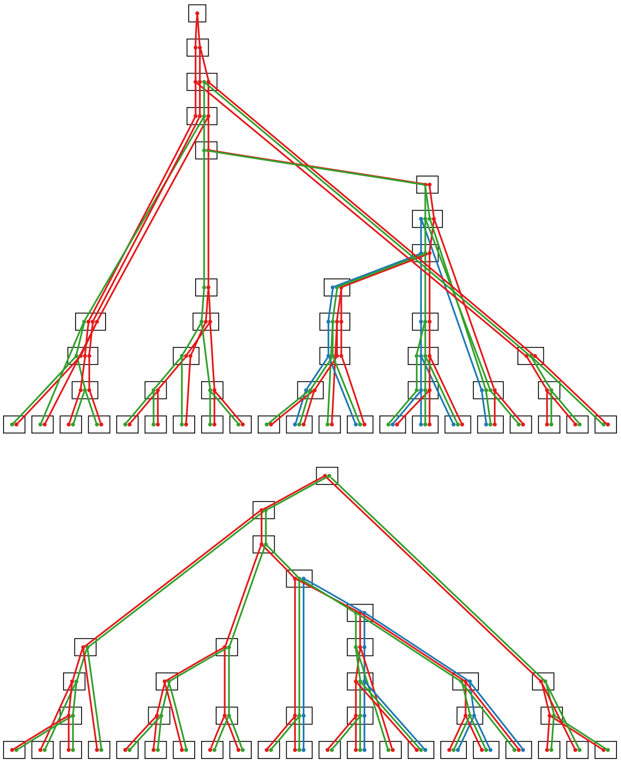


Fig. 10. Tree correction changes domain trees based on the plexus topology. The upper figure shows a plexus after merging. It clearly has a tree-like structure, which makes the distortions in the domain trees visible. Many genes show coordinated losses as examplified in Fig. 6, which induces false repeats especially at the root. The lower figure shows the same plexus after tree correction. The structure of the gene graph is kept, but all false repeat repeats and coordinated losses are eliminated.

Path Detachment: The number of unnecessary losses can be further reduced. Let there be any gene edge path $((g_1, g_2), (g_2, g_3))$. If all domain nodes in $\Delta(g_2)$ that have parents in g_1 (i. e. $\nu^-(g_2, g_1)$) only have children in g_3 (i. e. $\mathrm{N}^1(\nu^-(g_2, g_1)) \subseteq \Delta(g_3)$), then this induces unnecessary domain losses, as the composition g_2 is only supported by one direct successor. One can therefore split the gene node g_2 into $\nu^-(g_2, g_1)$ and $g_2 \setminus \nu^-(g_2, g_1)$, and apply this procedure recursively to their direct successors, thus reducing the number of loss events (see Fig. 9). Again, subsequent gene edge contraction maintains the integrity of the compact plexus.

Combining all of the above yields our final heuristic:

Algorithm 1: Heuristic for compact plexus reconstruction

```
plexus $P$, $d$, $t$
for $g_1, g_2 \in V(G)$ do
    $Q$.push($c(g_1, g_2, d, t)) : (g_1, g_2)$)
end for
while $Q$ has finite key do
    $(g_1, g_2) = Q$.pop()
    $g := g_1 \sqcup g_2$
    transitive reduction to $g$
    for $(g_1, g_2) \in A$ do
        $Q$.update($c(g_1, g_2, d, t) : (g_1, g_2)$)
    end for
end while
recursive tree correction from all roots
recursive path detachment from all roots
```

4.3 Time Complexity

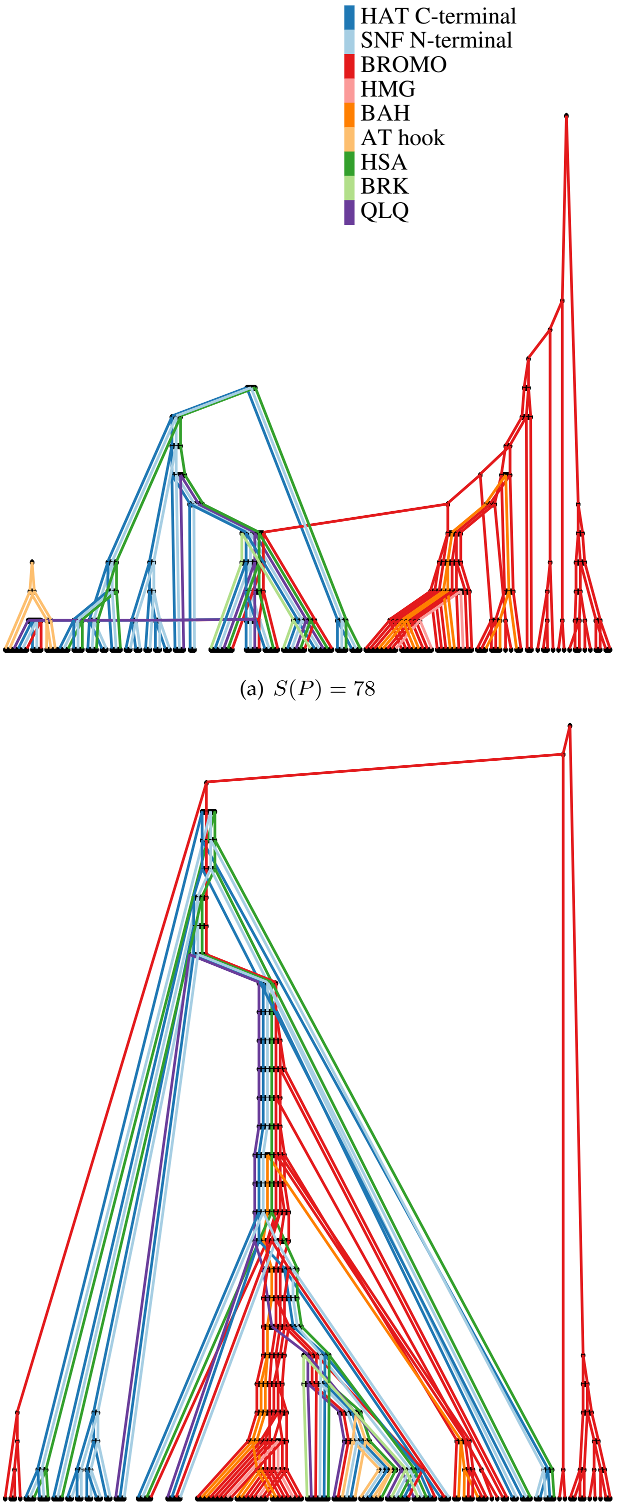
The merge step dominates the running time. To decide which gene nodes to merge, one has to calculate path distances between their direct successors. A plexus is a DAG, so shortest paths for all pairs can be calculated in $O\{|V|\,|E|\}$ [8]. Since initially the gene graph is a DAG for which $|E| \subset O\{|V|\}$, and the number of edges does not increase during gene merging, all-pairs shortest paths can be calculated in $O\{|V|^2\}$.

Finding the smallest d-compatibility by pairwise comparison takes time in $O\{|V|^2\}$. With L being the leaf set, $|L| < |V|$ is the number of profiles one has to check, so the time for finding the d-t-closest pair lies in $O\{|V|^2 + |L| \cdot |V|^2\} \subseteq O\{|V|^3\}$. As the number of gene nodes decreases with every merge, one has to perform this $O\{|V|\}$ times, if all distances are recalculated in each step. Hence the time complexity of the merge step is $O\{|V|^4\}$. Traversing the plexus for both tree correction and path detachment does not add any terms above that bound.

In terms of absolute running time, our results are preliminary. Our original proof-of-concept implementation took about 3 minutes for the larger examples in this paper. Besides implementation details, the long running time can be primarily attributed to recalculating all d-t-distances instead of using the optimization illustrated in Fig. 7. A faster version is currently under development, first tests showed an improvement to about 5 seconds.

5 Application

Our heuristic is relevant to protein domains that occur in different contexts. With thousands of available genomes and many of the promiscous domains occuring in dozens of copies, large domain trees can be



(a) $S(P) = 78$

(b) $S(P) = 131$

Fig. 11. Two compact plexūs of domains in histone acetyltransferase complexes. In (a) we used 1-reconcilability, tree correction and path detachment, resulting in a late fusion event of the BROMO domain. In (b) only d-compatibility was used. Labeled high-resolution figures can be downloaded for detailed analysis from http://genome.cs.iastate.edu/CBL/ISBRA10/thesis.zip

constructed. For this work, we chose examples that remain comprehensible to the reader and depictable in figures. Many multi-domain proteins are involved in transcriptional regulation, e. g. via chromatin remodeling, and we selected our examples from this context.

We initially ensured the results of the heuristic on previously published proven albeit simple examples [15] and obtained identical results (not shown). For input data with no errors or inconsistencies, several of the steps described above are unnecessary.

5.1 Construction of Phylogenetic Trees on Domains

To obtain typical input data, we started at the sequence level by selecting suitable domain combinations and species. Domains were identified with `hmmsearch` in the UniProt database [22], parsed and aligned with `hmmalign` from the HMMER package, version 3.0 [24]. Maximum Likelihood trees were constructed using `PhyML` under the VT model, four rate categories and estimated γ [26]. Notung 2.6 was used to root the domain trees [27] using duplications without losses and known species trees. Other methods to root the tree to remove the dependency on the species tree, such as outgroups or using the midpoint of the longest branch can be used alternatively.

5.2 BROMO Domains and Histone Acetyltransferases

We selected the proteins containing the BROMO, the N-terminal SNF2 and the C-terminal conserved helicase domains of the histone acetyltransferases in *H. sapiens*, *D. melanogaster*, *S. cerevisiae*, *S. pombe*, and *A. thaliana*. The BROMO domain is one of the most frequently recombining domains [5] and plays a crucial role in the recognition of acetylated lysines on histone tails. We also obtained trees for additional domains found with the three marker domains (see Fig. 11)

Our result obtained heuristically scored 78 (Fig. 11). The fusion of BROMO and SNF2 domains is the most important single event in the resulting plexus. It happens after duplication of ancestral SNF2 domain proteins and is not lost in any of its children. The heuristic obtains useful results even in the face of incomplete and erronous input data. The AT hook domain – more of a motif really – is difficult to identify and absent or incompletely identified in several proteins. Our solution depicts the presence of AT hooks in several proteins as independent fusion events. While it is rather probable that intervening sequences also carry the motif, this solution is satisifiable given the input data.

When t-reconcilability, path detachment and tree correction are not used, the results decrease in quality rapidly (Fig. 11(b)). The elements in the resulting trunk do not resemble the composition of extant MDPs and induce multiple losses. The BROMO domain fusion is placed at the top, which leads to many losses of the BROMO domain and thereby increases the score significantly. Genes in the central trunk are mostly mapped to the ophistokonts rather than the fungi or metazoa, which further increases the score.

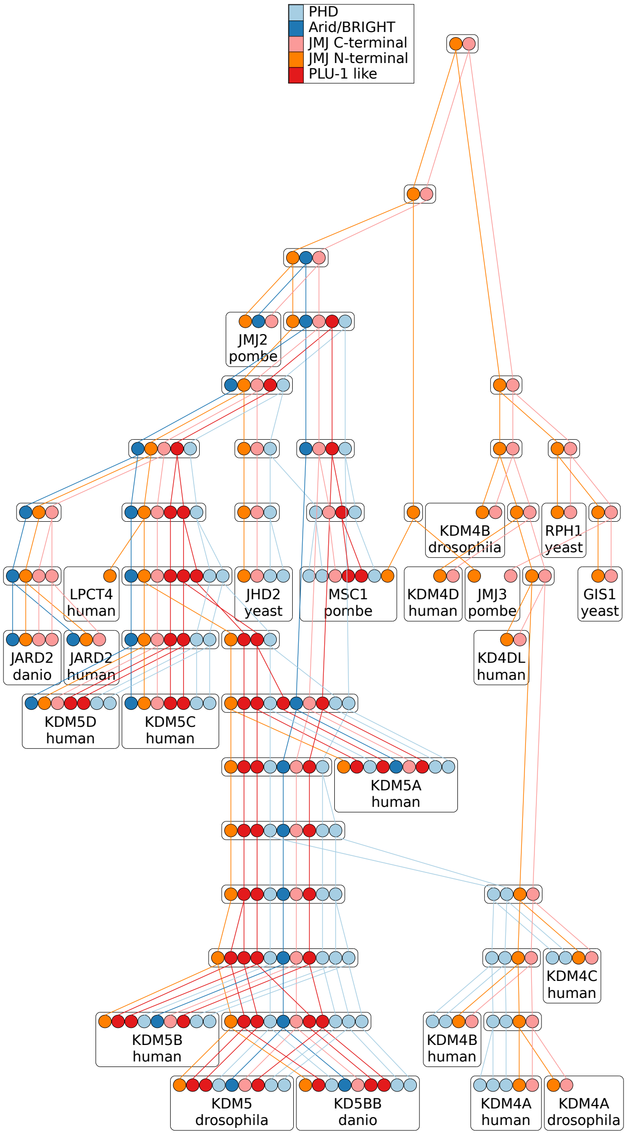


Fig. 12. Reconstruction of Jumonji-associated domains. The color of a tree denotes its domain family. Proteins are presented in rounded boxes, extant proteins are labeled. See 5.3 for in-depth explanations.

5.3 Jumonji-associated Domains

The Jumonji protein (jmj) is a histone demethylase, which regulates chromatin structure and is involved in embryonic development. It is often associated with additional promiscous domains such as ARID/BRIGHT or PHD [29]. We built trees as above on domains in proteins containing the N-terminal JMJ domain in *H. sapiens, D. melanogaster, S. cerevisiae, S. pombe*, and *D. rerio* (Fig. 12). The resulting plexus topology is determined by the N- and C-terminal JMJ domains, which are present in all extant proteins. Previous characterizations were based on the C-terminal domain only [29]. Problems in the input trees can be readily identified and the plexus can serve a test of congruence of the input trees. The JMJ3 proteins in *S. pombe* (center right) implausibly results from a fusion of the C-terminal and N-terminal domain, which would require the fission and subsequent fusion of the same domains. More likely, one of the domain trees contains a wrong split. Phylogenetic trees built only on one domain are blind to such discrepancies. Another noteworthy event in the plexus is the prediction that there were two events leading to proteins containing PHD and JMJ domains, a gain in the top of the main center main trunk and a split and fusion leading to the predecessors of KDM4A in human and Drosophila. While this might appear implausible at first, it should be noted that others have observed similar re-inventions of domain architectures using high-quality domain trees [23]. A survey of a larger body of possible fusion events of the PHD domain would be of interest.

6 Conclusion and Outlook

We introduced a graph-theoretic concept to model MDP evolution, derived an optimization problem and presented an approach to solve it. The application to real data infers credible scenarios for the evolution of MDPs and can reveal inconsistencies in the input trees. Further improvements to d-compatibility could enhance the use of real data, and extending it to weighted paths would allow the use of bootstrap-valued DAGs instead of trees to deal with the ambiguities in phylogenetic signals. It could also be modified to handle unrooted trees. Including domain order into the model and modifying the compatibility constraint would be helpful in separating true losses from missing annotations. There are several issues on how to determine a consensus order, especially when a domain has two related domains of another family on either side.

As shown in Fig. 11(b) in the heuristic, random compatibility can be a problem. Although we adress it by t-reconcilability, path detachment and tree correction, the development of a statistical model that assigns a p-value to a plexus topology would be worthwhile.

Additional properties could be exploited to improve a heuristic solution or find an approximation. The various concepts in our heuristic are in parts rooted in practical considerations, and their interdependencies are not explicitly stated. We would like to find a monolithic formulation and translate the plexus construction to a known algorithmic problem.

Constraint optimization approaches might allow for considerable speedup in the implementation and possibly find optimal solutions.

Acknowledgments

We thank Max Homilius, Ivan Kel, Christoph Standfuß and Sebastian Thieme for sharing data and code. This work was supported in part by the NSF AToL program DEB 0830012.

References

[1] Song, N., Joseph, J.M., Davis, G.B., Durand, D.: Sequence similarity network reveals common ancestry of multidomain proteins. PLoS Computational Biology **4**(5) (2008)

[2] Doolittle, R.F.: The multiplicity of domains in proteins. Annual Review of Biochemistry **54** (1995) 287–314

[3] Koonin, E.V., Altschul, S.F., Bork, P.: BRCA1 protein products... Functional motifs... Nature Genetics **13**(3) (1996) 266

[4] Ekman, D., Björklund, Å.K., Frey-Skött, J., Elofsson, A.: Multi-domain Proteins in the Three Kingdoms of Life: Orphan Domains and Other Unassigned Regions. Journal of Molecular Biology **348**(1) (April 2005) 231–243

[5] Basu, M.K., Carmel, L., Rogozin, I.B., Koonin, E.V.: Evolution of protein domain promiscuity in eukaryotes. Genome Res. (2008) gr.6943508+

[6] Apic, G., Gough, J., Teichmann, S.A.: An insight into domain combinations. Bioinformatics **17 Suppl 1** (2001)

[7] Kummerfeld, S.K., Teichmann, S.A.: Relative rates of gene fusion and fission in multi-domain proteins. Trends in Genetics **21** (2005)

[8] Duin, C. W.: Two fast algorithms for all-pairs shortest paths. Computers & Operations Research **34**(9) (2007) 2824–2839

[9] Yang, S., Doolittle, R.F., Bourne, P.E.: Phylogeny determined by protein domain content. Proceedings of the National Academy of Sciences **102**(2) (2005) 373–378

[10] Björklund, Å.K., Ekman, D., Light, S., Frey-Skött, J., Elofsson, A.: Domain Rearrangements in Protein Evolution. Journal of Molecular Biology **353**(4) (November 2005) 911–923

[11] Przytycka, T., Davis, G., Song, N., Durand, D.: Graph Theoretical Insights into Evolution of Multidomain Proteins. Journal of Computational Biology **13**(2) (2006) 351–363

[12] Fong, J.H., Geer, L.Y., Panchenko, A.R., Bryant, S.H.: Modeling the Evolution of Protein Domain Architectures Using Maximum Parsimony. Journal of Molecular Biology **366**(1) (February 2007) 307–315

[13] Ciccarelli, F.D., von Mering, C., Suyama, M., Harrington, E.D., Izaurralde, E., Bork, P.: Complex genomic rearrangements lead to novel primate gene function. Genome Research **15**(3) (2005) 343–351

[14] Lucas, J.I., Arnau, V., Marín, I.: Comparative genomics and protein domain graph analyses link ubiquitination and RNA metabolism. Journal of Molecular Biology **357**(1) (2006) 9–17

[15] Behzadi, B., Vingron, M.: Reconstructing Domain Compositions of Ancestral Multi-domain Proteins. Springer, Berlin/Heidelberg (2006) 1–10

[16] Huson, D.H., Bryant, D.: Application of phylogenetic networks in evolutionary studies. Molecular Biology and Evolution **23**(2) (2006) 254–67

[17] Moret, B.M.E., Nakhleh, L., Warnow, T., Linder, C.R., Tholse, A., Padolina, A., Sun, J., Timme, R.: Phylogenetic Networks: Modeling, Reconstructibility, and Accuracy. IEEE/ACM Transactions on Computational Biology and Bioinformatics **1**(1) (2004) 1–12

[18] Moore, A.D., Björklund, Å.K., Ekman, D., Bornberg-Bauer, E., Elofsson, A.: Arrangements in the modular evolution of proteins. Trends in Biochemical Sciences **33**(9) (2008) 444–451

[19] Snel, B., Bork, P., Huynen, M.: Genome evolution: gene fusion versus gene fission. Trends in Genetics **16**(1) (2006) 9–11

[20] Wang, W., Yu, H., Long, M.: Duplication-degeneration as a mechanism of gene fission and the origin of new genes in Drosophila species. Nature Genetics **36**(5) (May 2004) 523–7

[21] Wiedenhoeft, J.: Phylogenetic Reconstruction of Ancestral Multidomain Proteins (2009) BSc thesis. http://genome.cs.iastate.edu/CBL/ISBRA10/thesis.zip

[22] UniProt Consortium.: The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Res. (2010) **38** (Database issue) D142-8.

[23] Forslund, K., Henricson, A., Hollich, V., Sonnhammer, E.L. Domain tree-based analysis of protein architecture evolution Molecular Biology and Evolution (2008) **25**(2) 254-64

[24] Finn, R.D., Tate, J., Mistry, J., Coggill, P.C., Sammut, S.J., Hotz, H.R., Ceric, G., Forslund, K., Eddy, S.R., Sonnhammer, E.L.L., Bateman, A.: The Pfam protein families database. Nucleic Acids Research **36**(Database issue) (January 2008) D281–8

[25] Marija Buljan, M., Frankish, A., Bateman, A. Quantifying the mechanisms of domain gain in animal proteins Genome Biology **11** (2010) R74

[26] Guindon, S., Gascuel, O.: A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic Biology **52**(5) (October 2003) 696–704

[27] Durand, D., Halldorsson, B.V., Vernot, B.: A hybrid micro-macroevolutionary approach to gene tree reconstruction. Journal of Computational Biology **13**(2) (2006) 320–335

[28] Boxem, M., et al.: A Protein Domain-Based Interactome Network for C. elegans Early Embryogenesis. Cell **134**(3) (2008) 534–545

[29] Takeuchi T, Watanabe Y, Takano-Shimizu T, Kondo S.: Roles of jumonji and jumonji family genes in chromatin regulation and development Developmental Dynamics (2006) **235**(9) 2449–59.

[30] Jin, J., et al. : Eukaryotic protein domains as functional units of cellular evolution Science Signaling **98**(2) (2009) ra76

John Wiedenhoeft studied ethnomusicology and communications research in Berlin, and received his B.Sc. in bioinformatics at Free University of Berlin in 2009, where he currently pursues his M.Sc.

Roland Krause received his undergraduate education in biotechnology from the Applied University Mannheim and his doctorate in biochemistry from the University of Heidelberg in 2004 for the analysis of protein complexes. His research interests are mechanisms of gene regulation and their evolution.

Oliver Eulenstein studied computer science at Paderborn and Bonn Universities in Germany and received the PhD degree in 1998. From 1998 to 2000 he held a postdoctoral position with Dan Gusfield at the University of California Davis. He is currently an associate professor of computer science at Iowa State University. His research interests focuses on computational biology; particularly on developing algorithms for addressing challenging computational problems in phylogenetics.