Inferring Evolutionary Scenarios for Protein Domain Compositions

John Wiedenhoeft[1], Roland Krause[1,2], and Oliver Eulenstein[3]

[1] Free University of Berlin
john.wiedenhoeft@fu-berlin.de
[2] Max Planck Institute for Molecular Genetics Berlin
roland.krause@molgen.mpg.de
[3] Iowa State University
oeulenst@cs.iastate.edu

Abstract. Essential cellular processes are controlled by functional interactions of protein domains, which can be inferred from their evolutionary histories. Methods to reconstruct these histories are challenged by the complexity of reconstructing macroevolutionary events. In this work we model these events using a novel network-like structure that represents the evolution of domain combinations, called plexus. We describe an algorithm to find a plexus that represents the evolution of a given collection of domain histories as phylogenetic trees with the minimum number of macroevolutionary events, and demonstrate its effectiveness in practice.

1 Introduction

Inferring the evolutionary history of domain compositions of proteins is a key problem for the elucidation of protein function from large-scale genomic data. In essence, a *domain* is an independent and evolutionary mobile sub-unit of a protein [1]. The recognition of such characteristics has led to breakthroughs in the determination of protein function, e. g. for the oncogene BRCA1 [2]. The vast majority of proteins in the higher Eukaryotes consist of several domains [3]. About 200 of these domains combine frequently into a rich variety of *multi-domain proteins* (MDPs) that are involved in essential cellular processes, including chromatin remodeling and signal transduction [4]. Recombination events of domains lead to similarities between proteins that have more than one common ancestor, and which are therefore not strictly homologous. These proteins can pose a major problem for phylogenetic inference in protein families [5].

Here, we describe a novel approach to reconstruct evolutionary MDP scenarios for which standard phylogenetic inference methods may not be appropriate. We formulate the *MDP evolution problem*, describe an effective heuristic to solve it and show that its implementation performs well in practice.

1.1 Background

After the discovery of mobile domain combinations in the 1980s, it required complete eukaryotic genome sequences for thorough investigations of the

M. Borodovsky et al. (Eds.): ISBRA 2010, LNBI 6053, pp. 179–190, 2010.
© Springer-Verlag Berlin Heidelberg 2010

phenomenon [1]. Genome wide studies of multi-domain proteins either utilize the order of the domains or study the co-occurence, but 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 [7,8]. [9] used a parsimony-based approach and simplified gene fusion, domain shuffling and retrotransposition events into tractable merge and deletion operations. [10] constructed a more elaborate model with 3 subclasses of fusion events for multi-domain proteins to reconstruct domain trees.

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 [11,12].

[13] suggested an approach incorporating domain histories to reconstruct ancestral domain compositions from a given collection of domain trees and a 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. The 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-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 interesting approaches were used with success for phylogenetic displays and mapping of events, reviewed in [14]. Our approach relates to [15], 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 Contribution of This Work

Our formalization of the *MDP evolution problem* is: given a collection of phylogenetic trees of extant domains, find scenarios that minimize the change in MDP composition. We describe an effective heuristic for this reconstruction problem and show that its implementation performs well in practice for a selection of proteins with frequently recombining domains. We do not rely on a given species tree but present a novel graph-theoretic network, called *plexus*, that allows to describe scenarios for the evolution of a collection of domain trees. We introduce three different instances of this network (see Fig. 1). The *expanded* plexus corresponds to a biological scenario, the *reconstructable* to what is obtainable from the phylogenetic reconstruction and the *compact* to a computationally feasible model.

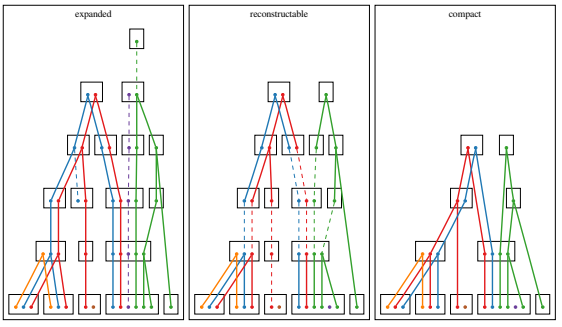


Fig. 1. Counterparts of plexūs for a set of MDPs and their domain trees. Edges mark inheritance relations between domains (dots) in MDPs (rectangles), different domain families are in different shades. The expanded plexus consists of evolutionary events only (see Fig.2). Its non-reconstructable edges are dashed and disappear in the reconstructable counterpart. Some of the resulting blocks may then contain only nodes with out-degree 1 (dashed). Contracting their out-edges results in the compact counterpart.

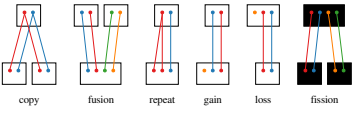


Fig. 2. Basic events in MDP evolution. Fission is modeled with elementary events (see Fig. 3).

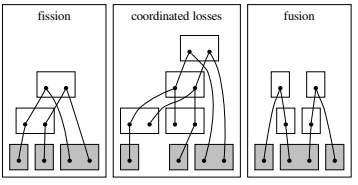


Fig. 3. Ambivalent explanations of fission events for two gene trees and three proteins

2 A Model for the Evolution of MDPs

To reconstruct MDP evolution, we require the composition of extant proteins and the phylogenetic relationships between domains of the same family. We now give an overview of the types of operations on MDPs and derive our model.

2.1 Evolutionary Events

We consider five macroevolutionary events (Fig. 2). Duplications and speciations are undistiguished *copy* events, as we do not rely on a species tree. *Fusion* describes the union of two ancestral MDP compositions via loss of terminal and initial segments or translocation. *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 [16]. A *gain* is the introduction of the root of a domain tree and 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 [17]. An alternative scenario for fission involves gene duplication with subsequent coordinated domain losses [18]. A third variant explains the observations by a fusion process (see Fig. 3). We model fission by a combination of other basic events and the score of the optimal plexus topology happens to be invariant to the explicit series of events.

2.2 The Plexus

Generally, a plexus is a meshwork of branching and rejoining strands, e. g. a network of blood vessels or neurons as in choroid plexus or solar plexus. It connotates that the strands have a direction as in blood flow or action potential propagation. We use the term to describe the aggregation and segregation of phylogenetic domain trees such that the nodes correspond to extant and reconstructed ancestral MDPs. We assume domain trees to be rooted and fully binary for this work.

An expanded plexus is constructed by linking the basic evolutionary events (see Fig. 1, left side and Fig. 2). This results in a *directed acyclic graph* (DAG) whose nodes are sets of domain tree nodes and whose edges are made up by the edges of the domain trees; to avoid confusion, we call the plexus' nodes *blocks* and its edges *arcs*. As shown in Fig. 1, the trees in the plexus are *not* necessarily the input trees but display them. Any subtree that contains no node in the plexus' leaves will not be in a subgraph of the input trees. Also, any root that has only one child will not be found in the reconstructed domain trees. The dashed lines in Fig. 1 (left side) represent such non-reconstructable edges. There is an infinite number of plexūs with identical compact counterparts, thus displaying the same input domain trees. The only plexus we can actually reconstruct is the one we obtain by deleting the non-reconstructable subtrees from the expanded plexus by removing any nodes in non-leaf blocks that have a combined in- and out-degree ≤ 1. On this reconstructable plexus, we can apply a scoring scheme that approximates the number of MDP evolution events.

The reconstructable plexus is still not very handy as it has infinitely many possible topologies. We can however restrict the topologies to a finite number by requiring that each block must contain at least one node with out-degree 2, which makes the number of tree nodes an upper bound to the number of blocks. By contracting out-arcs of blocks containing only nodes of out-degree 1, we transform a reconstructable into a compact plexus. In Fig. 1 (middle), these are the arcs made up by the dashed edges.

With the definition of the compact plexus, the problem is reduced to partitioning domain tree nodes. It is 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 DAG. 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 $u, w \in V$ and $j \in \mathbb{Z}^+ \cup \{\infty\}$ we (i) write $u \sim_j w$, if $u \neq w$ and there is a path from u to w of at most j edges in G, and (ii) define

$u \sim_{-j} w := w \sim_j u$. If $u \sim_k w$ and $k > 1$, then we call u a *predecessor* of w, and w a *successor* of u. In the case $k = 1$, we use the terms *direct predecessor* and *direct successor* accordingly. We say that u and w are *connected* if $u \sim_\infty w$. 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 $a \to b \to c$, $N^{1,1}\{a\} = N^1\{b\} = \{c\}$.

3.2 Plexus and Evolutionary Events

Let $G := (V, E)$ be a DAG. We call the graph $P(G) := (\mathcal{V}, \mathcal{E})$ a *plexus over* G if the following conditions are satisfied: (i) $P(G)$ is a DAG, (ii) $\mathcal{V}$ is a partition of V such that no pair of nodes in any $\mathcal{v} \in \mathcal{V}$ is connected in G, and (iii) $(\mathcal{a}, \mathcal{b}) \in \mathcal{E}$ iff there is a pair of nodes $a \in \mathcal{a}$ and $b \in \mathcal{b}$ for which $(a, b) \in E$.

We refer to plexus vertices as *blocks* and edges between blocks as *arcs*. Plexus notation is identical to graph notation, but is distinguished for clarity by *calligraphic script*. Blocks represent the composition of an MDP and arcs describe their inheritance relations. $\mathrm{C}_{\mathcal{w}}(\mathcal{v}) := \{p | p \in \mathcal{v} : \exists c \in \mathcal{w} : (p, c) \in E(G)\}$ is the set of nodes in block $\mathcal{v}$ that have children in block $\mathcal{w}$, and $\mathrm{P}_{\mathcal{v}}(\mathcal{w}) := \{c | c \in \mathcal{w} : \exists p \in \mathcal{v} : (p, c) \in E(G)\}$ the set of nodes in $\mathcal{w}$ with parents in $\mathcal{v}$.

Let $P := (\mathcal{V}, \mathcal{E})$ be a plexus. We call P *expanded* if for each of its blocks $\mathcal{b} \in \mathcal{V}$ either $(\mathcal{b}, \mathrm{N}^1\{\mathcal{b}\})$ or $(\mathrm{N}^{-1}\{\mathcal{b}\}, \mathcal{b})$ is an MDP evolution event. For brevity, a formal definition of MDP evolution events is omitted here but can be found in [19]. P is called *reconstructable* if no non-terminal block contains any node for which the sum of its in- and out-degree is less then 2 . A reconstructable plexus 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 an in- or out-degree of 0 and their incident edges. Let $\mathcal{e} := (\mathcal{v}, \mathcal{w}) \in \mathcal{E}$ be an arc such that $\forall v \in \mathrm{C}_{\mathcal{w}}(\mathcal{v}) : \deg^+(v) = 1$ and $\forall w \in \mathrm{P}_{\mathcal{v}}(\mathcal{w}) : \deg^-(w) = 1$. Let $\mathcal{e}$ be an arc such that $\forall e := (v, w) \in \mathcal{e} : \deg^+(v) = 1, \deg^-(w) = 1$. The operation of contracting all $e \in \mathcal{e}$ and merging $\mathcal{v}$ with $\mathcal{w}$ is called *arc contraction*. A plexus C is said to be *contracted* if it contains no contractible arcs, and *contracted counterpart* of a plexus P if it is contracted and can be obtained by contracting arcs in P. This is similar to the concept of *minors* in undirected graphs. A contracted plexus C is called the *compact counterpart* of a plexus P, if there is a reconstructable counterpart R of P such that C is a contracted counterpart of 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 blocks connected by an arc contain nodes that are not related to any node in the other block. In contrast to copy and repeat, the direct successor blocks 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 arcs in fusion, gain and loss, and hence to exactly those events that we consider to be of evolutionary importance. We can reconstruct them from the compact counterpart.

The number of losses accounting for a block v from all its ancestors is $\sum_{p\in \mathrm{N}^{-1}\{v\}} \left(|p| - |\mathrm{P}_p(v)|\right)$ which equals $\sum_{p\in \mathrm{N}^{-1}\{v\}} |p| - \sum_{p\in \mathrm{N}^{-1}\{v\}} |\mathrm{P}_p(v)|$.

The number of gains is equal to the number of domain trees and constant for all topologies, and therefore omitted. The remaining problem is to address the contraction of fusion arcs, which can increase the in-degree of a block. We can relate the number of fusion arcs in a plexus to its compact counterpart. Let P_E be an expanded plexus and P_C its compact counterpart. Then the number of fusion arcs in P_E equals $\sum_{b\in \mathcal{V}(P_C)} \max\{0, 2\cdot \deg^-(b) - 2\}$.

The order of fusions is lost during compactification but the number of domain changes depends on that order. Consider a block with in-degree 3 and predecessor blocks of size 1, 2 and 3. Combining 1 and 2 first creates an out-arc of size 3, and then merging with the third block creates an out-arc of size 6. In contrast, combining 2 and 3 first produces out-arcs 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. As the real sequence of fusions is unknown, we use the mean number of nodes at the end of in-arcs, which defines the following *fusion cost* $\frac{\max\{2\cdot\deg^-(v)-2,0\}}{\max\{\deg^-(v),1\}} \cdot \sum_{p\in \mathrm{N}^{-1}\{v\}} |\mathrm{P}_p(v)|$.

This formula also holds in cases in which an arc involves a tandem repeat. Combining the above equations for losses, gains and fusions and summing up over all blocks yields the *plexus score* $S(P)$ as the score of its compact counterpart P_C as

$$S(P) = \sum_{v\in \mathcal{V}(P_C)} \sum_{p\in \mathrm{N}^{-1}\{v\}} \left(|p| + \left(1 - \frac{2}{\deg^-(v)}\right) \cdot |\mathrm{P}_p(v)| \right).$$

Note that $\max\{\ldots\}$ in the fusion cost formula only serves to avoid negative costs for root blocks. As the index set $\mathrm{N}^{-1}\{v\}$ is empty in this problematic case, the formula is simplified.

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

Problem 1 (plexus reconstruction)
Instance: A set T of fully binary domain trees and a partition $\mathcal{L}$ of their combined leaf set such that each set block corresponds to a known MDP composition.
Find: a compact plexus P in which $\mathcal{L}$ is the leaf block set and which displays T such that the plexus score $S(P)$ is minimal.

4 Heuristic

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.

Our method works in three steps, which correspond to the events we want to minimize. In the initial *block merging* step, we merge non-leaf blocks according to a compatibility criterion that asserts an out-degree ≤ 2 (d-compatibility), and allows only for compositions that resemble those of the input (t-reconcilability). The latter also tolerates compositions that are close to the observed with additional domains to account for false tree splits but mainly reduces the number of *fusions*. In the second step (*tree correction*), we attempt to correct the placement of tree nodes based on the preliminary topology of the plexus to minimize the number of *coordinated losses*, i. e. two domains of the same family that each have only one child, but in different direct successor blocks, as shown for the solid domain in Fig. 4. In the final *path detachment* step, we separate sub-blocks consisting of nodes that are placed too high in the plexus by previous steps, which reduces *unnecessary losses*.

4.1 Block Merging

To reduce the number of fusion events we merge non-leaf blocks. To restrict the merged blocks' out-degree to ≤ 2, we use transitive reduction of arcs. An arc (v, w) is a *transitive arc* (v, w) if $w \in \mathrm{N}^k\{v\}$ for any $k > 1$. The path of a transitive arc (v, w) is given by $(v, b_1, \ldots, b_k, w)$. One can insert k nodes in each edge $e = (v_v, v_w)$ in (v, w), thus creating paths $(v_v, v_1, \ldots, v_k, v_w)$. Placing each v_i into b_i reduces the out-degree of v by 1. Consequently, blocks are *reducible* if their out-degree can be reduced by transitive reduction of outgoing arcs. This holds iff $\mathrm{N}^1\{b\} \cap \mathrm{N}^{1,\infty}\{b\} \neq \emptyset$.

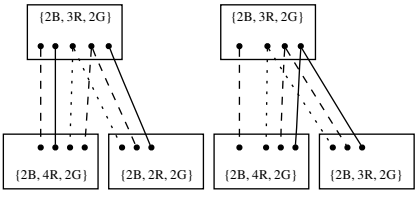


Fig. 4. Composition profiles. In the reconstruction of an ancestral block we show a typical artifact from errors in the phylogenetic reconstruction that leads to additional domains in the ancestral block, here the nodes of the tree with solid edges in the left figure. Despite the left variant containing two copies of the solid domain family the outdegree-profile for both variants is identical.

For any pair of blocks it is necessary to know whether there is a transitive reduction to the merged block such that its out-degree does not exceed 2. Assume that the blocks are irreducible, as we could always apply transitive reduction before a merge.

Theorem 1 (minimal out-degree). *Let v and w be two irreducible blocks such that $v \notin \mathrm{N}^{\infty}\{w\} \cup \mathrm{N}^{-\infty}\{w\}$. Let χ be a block obtained by merging v and w. Then the minimal out-degree $\deg^{\triangleleft}(\chi)$ that can be obtained by a sequence of transitive reductions to χ is* $\deg^{\triangleleft}(\chi) := \left|\mathrm{N}^1\{v\}\right| + \left|\mathrm{N}^1\{w\}\right| + \left|\mathrm{N}^1\{v\} \cap \mathrm{N}^1\{w\}\right| - \left|\mathrm{N}^1\{v\} \cap \mathrm{N}^{\infty}\{w\}\right| - \left|\mathrm{N}^{\infty}\{v\} \cap \mathrm{N}^1\{w\}\right|$.

A proof is omitted for brevity and can be found in [19]. We can find all pairs of blocks that can be merged without violating neither out-degree d nor compact plexus properties by the following criterion:

Definition 1 (d-compatibility). *Two irreducible blocks v and w are called* d-compatible *if $\deg^{\triangleleft}(v \cup w) \leq d$, i. e. one can obtain a block with an out-degree of at most d by merging v and w and applying a sequence of transitive reductions to the merged block. However, if either of them is a leaf, or $v \notin \mathrm{N}^{\infty}\{w\} \cup \mathrm{N}^{-\infty}\{w\}$ (blocks are related), then they are incompatible.*

Definition 2 (reduction cost r). *Let v be a block with an out-degree greater than 0. The* reduction cost *of an outgoing arc pointing to block $w \in \mathrm{N}^1\{v\}$ is the smallest $k > 0$ for which $w \in \mathrm{N}^k(\mathrm{N}^1\{v\} \setminus \{w\})$, or 0 if there is no such k, i. e. the arc is not transitive and thus cannot be removed. The reduction cost $r(v)$ of a block is the sum of costs of all its outgoing arcs.*

An *a-priori* set of candidates excluding all blocks that cannot be compatible with the current block ensures a tractable solution space. Only the direct predecessors of all successors of each block v need to be checked for compatibility.

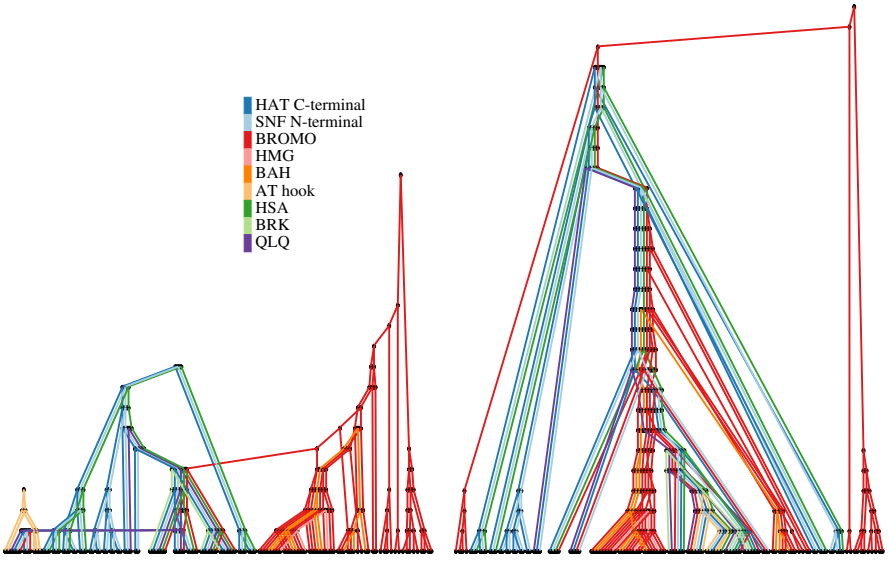
2-compatibility alone leads to domain compositions that do not resemble recent MDPs, leading to many losses as seen in Fig. 5(b). Many compatibilities arise merely by chance or by false tree splits. We therefore ensure that blocks *resemble recent compositions* by the following:

Definition 3 (composition profile). *Let $M = \{d_1, \ldots, d_k\}$ be a set of nodes in a block. M is partitioned into subsets $\{F_1, \ldots, F_m\}$ of nodes that belong to the same input tree, the set of families is denoted by representants $p := \{[\![F_1]\!], \ldots, [\![F_m]\!]\}$. Let $m(\cdot) : [\![F_i]\!] \to \mathbb{N}$ be the mapping $m([\![F_i]\!]) = 2 \cdot |\{n | n \in F_i, \deg^{+}(n) = 0\}| + \sum_{d \in F_i} \deg^{+}(d)$. Then (p, m) is called the* composition profile *of M.*

Definition 4 (t-reconcilability). *A profile p_1 is called* t-reconcilable *to a profile p_2 if $\forall [\![F_i]\!]_1 \in p_1 : \exists [\![F_i]\!]_2 \in p_2 : [\![F_i]\!]_1 = [\![F_i]\!]_2, m([\![F_i]\!]_1) \leq m([\![F_i]\!]_2) + t$, 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 block 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. 4: on the right the upper block resembles its left direct successor, whereas on the left it contains one solid domain more than any of its direct successors.

Both predecessor blocks have the same profile though, since both solid domains have only one out-edge each. We might call this a *coordinated loss* of the solid domain; this will either be caused by a tree root being placed in the block above, but will often occur due to false tree splits. These might introduce disruptions to the optimal topology. t-reconcilability aims to compensate this, while providing a concept of similarity to recent compositions. One should choose t to be small to avoid meaningless ancestral compositions and large loss counts, but $t = 0$ assumes that the topologies of all input trees are correct, which will rarely be the case. $t = 1$ yielded the best results in our hands. Combining d-compatibility and t-reconcilability provides us with a criterion for the merges to prefer and to avoid:



(a) $S(P) = 45$ (b) $S(P) = 61$

Fig. 5. Two compact plexūs of domains in histone acetyltransferase complexes. In (a) we used 1-reconcilability, tree correction and path detachment, predicting a late fusion event of the BROMO domain. In (b) only d-compatibility was used, resulting in a single-source plexus with multiple losses and compositions that do not resemble extant MDPs. Note that, among others, the BROMO domain fusion is placed much too high (horizontal edge at top). Labeled high-resolution figures can be downloaded for detailed analysis from `http://genome.cs.iastate.edu/CBL/ISBRA10/thesis.zip`

Definition 5 (d-t-distance). *If two blocks v, w are d-compatible and the profile of $v \cup w$ is t-reconcilable to a profile of any input composition, their* d-t-distance $c(v, w)$ *is $r(v \cup w)$, otherwise it is ∞.*

Initially, we alternate between transitive reduction of all blocks and merging the two blocks with the shortest d-t-distance, until there is no pair whose distance is $< \infty$. We avoid merging repeat blocks with copy blocks and thus violating

compact plexus properties by inserting an additional block in the copy block's out-arc and merging it with the repeat block.

4.2 Tree Correction

The above procedure can introduce new blocks below old ones, thus pushing some tree nodes higher during block merges, thus *stretching* subtrees and give rise to additional losses. To compensate for this, we introduce *tree correction*: two tree nodes can be merged if they are in the same block, have the same parent node, and the out-degree of the merged node is ≤ 2 or can be reduced by recursively merging child nodes respectively. Root nodes must not be merged. If the common parent of two nodes being merged is a root node, it can be deleted after the merge, as the merged node will be a new root.

4.3 Path Detachment

Let there be any arc path $((a, b), (b, c))$. If all nodes in b that have parents in a (i. e. $\mathrm{P}_a(b)$) only have children in c (i. e. $\mathrm{N}^1(\mathrm{P}_a(b)) \subseteq c$), then this induces unnecessary domain losses, as the composition b is only supported by one direct successor. One can therefore split the block b into $\mathrm{P}_a(b)$ and $b \setminus \mathrm{P}_a(b)$, and apply this procedure recursively to their direct successors, thus reducing the number of loss events. After that, applying arc contraction ensures a compact plexus.

4.4 Time Complexity

The merge step dominates the running time. To decide which blocks to merge, one has to calculate path distances between their direct successors. A plexus is a DAG with all arcs having the same weight. Shortest paths are thus subgraphs of a breadth-first search tree. One has to create such a tree $|\mathcal{R}|$ times with $\mathcal{R}$ being the set of root blocks, so the time complexity of finding all pairs shortest paths is in $O\{|\mathcal{R}| \cdot (|\mathcal{V}| + |\mathcal{E}|)\}$. Since any block has two out-arcs at most, this is in $O\{|\mathcal{R}| \cdot |\mathcal{V}|\}$. Finding the smallest d-compatibility by pairwise comparison takes time in $O\{|\mathcal{V}|^2\}$. With $\mathcal{L}$ being the leaf set, $|\mathcal{L}|$ is the number of profiles one has to check, so the time for finding the d-t-closest pair lies in $O\{|\mathcal{R}| \cdot |\mathcal{V}| + |\mathcal{L}| \cdot |\mathcal{V}|^2\}$. As the number of blocks decreases with every merge, one has to perform this $\leq |\mathcal{V}|$ times at most, if all distances are recalculated in each step. The time complexity of the merge step is $O\left\{\sum_{v=1}^{|\mathcal{V}|} \left(|\mathcal{R}| \cdot v + |\mathcal{L}| \cdot v^2\right)\right\} \subseteq O\left\{\sum_{v=1}^{|\mathcal{V}|} v^3\right\} \subseteq O\{|\mathcal{V}|^4\}$ Both tree correction and path detachment traverses subtrees of the plexus but this is linear and depends on the number of blocks.

5 Application

We obtained identical results for the examples given in [13] (data not shown). To test our heuristic on proteins assembling to histone acetelyase complexes in *H. sapiens, D. melanogaster, S. cerevisiae, S. pombe*, and *A. thaliana*, we selected the proteins containing the BROMO, the N-terminal SNF2 and the

C-terminal conserved helicase domains of the histone acetyltransferases as identified in PFAM [20]. Domains were aligned with `hmmalign` of the HMMer package. Maximum Likelihood trees were constructed using `PhyML` [21]. Notung 2.6 was used to root the domain trees [22]. The input plexus had a score of 166, the result obtained heuristically scored 45 (Fig. 5(a)).

6 Conclusion and Outlook

We have presented an approach to reconstruct ancestral multi-domain proteins using plexūs. A suitable scoring scheme together with a heuristic allows finding near-optimal solutions.

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 ambiguity in the phylogenetic signal. It could also be modified to handle non-binary or unrooted trees. A compatibility constraint based on domain order would be helpful in separating true losses from missing annotations.

As seen in the heuristic, random compatibility is an important issue. We adress it by t-reconcilability, path-detachment and tree correction, but the development of a statistical model that assigns a p-value to a plexus topology would be worthwhile. Constraint optimization approaches might allow for considerable speedup in the implementation and possibly even find an optimal solution.

Acknowledgements

We thank M. Homilius, I. Kel, C. Standfuß and S. Thieme for sharing data. This work was supported in part by the NSF AToL program DEB 0830012.

References

1. Doolittle, R.F.: The multiplicity of domains in proteins. Annual Review of Biochemistry 54, 287–314 (1995)
2. Koonin, E.V., Altschul, S.F., Bork, P.: BRCA1 protein products... Functional motifs... Nature genetics 13(3), 266 (1996)
3. 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), 231–243 (2005)
4. Basu, M.K., Carmel, L., Rogozin, I.B., Koonin, E.V.: Evolution of protein domain promiscuity in eukaryotes. Genome Res. (2008) gr.6943508+
5. 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)
6. Apic, G., Gough, J., Teichmann, S.A.: An insight into domain combinations. Bioinformatics 17(suppl. 1) (2001)
7. Yang, S., Doolittle, R.F., Bourne, P.E.: Phylogeny determined by protein domain content. Proceedings of the National Academy of Sciences 102(2), 373–378 (2005)

8. Björklund, Å.K., Ekman, D., Light, S., Frey-Skött, J., Elofsson, A.: Domain Rearrangements in Protein Evolution. Journal of Molecular Biology 353(4), 911–923 (2005)
9. Przytycka, T., Davis, G., Song, N., Durand, D.: Graph Theoretical Insights into Evolution of Multidomain Proteins. Journal of Computational Biology 13(2), 351–363 (2006)
10. 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), 307–315 (2007)
11. 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), 343–351 (2005)
12. Lucas, J.I., Arnau, V., Marín, I.: Comparative genomics and protein domain graph analyses link ubiquitination and RNA metabolism. J. Mol. Biol. 357(1), 9–17 (2006)
13. Behzadi, B., Vingron, M.: Reconstructing Domain Compositions of Ancestral Multi-domain Proteins, pp. 1–10. Springer, Heidelberg (2006)
14. Huson, D.H., Bryant, D.: Application of phylogenetic networks in evolutionary studies. Molecular Biology and Evolution 23(2), 254–267 (2006)
15. 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), 1–12 (2004)
16. 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), 444–451 (2008)
17. Snel, B., Bork, P., Huynen, M.: Genome evolution: gene fusion versus gene fission. Trends in Genetics 16(1), 9–11 (2006)
18. 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), 523–527 (2004)
19. Wiedenhoeft, J.: Phylogenetic Reconstruction of Ancestral Multidomain Proteins (2009), BSc thesis, `http://genome.cs.iastate.edu/CBL/ISBRA10/thesis.zip`
20. 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), D281–D288 (2008)
21. Guindon, S., Gascuel, O.: A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic biology 52(5), 696–704 (2003)
22. Durand, D., Halldorsson, B.V., Vernot, B.: A hybrid micro-macroevolutionary approach to gene tree reconstruction. Journal of Computational Biology 13(2), 320–335 (2006)