Stolzer et al. BMC Bioinformatics 2015, 16(Suppl 14):S8
http://www.biomedcentral.com/1471-2105/16/S14/S8

BMC Bioinformatics

RESEARCH Open Access

Event inference in multidomain families with phylogenetic reconciliation

Maureen Stolzer[1], Katherine Siewert[1,2], Han Lai[1], Minli Xu[1], Dannie Durand[1,3*]

From 13th Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics
Frankfurt, Germany. 4-7 October 2015

Abstract

Background: Reconstructing evolution provides valuable insights into the processes of gene evolution and function. However, while there have been great advances in algorithms and software to reconstruct the history of gene families, these tools do not model the domain shuffling events (domain duplication, insertion, transfer, and deletion) that drive the evolution of multidomain protein families. Protein evolution through domain shuffling events allows for rapid exploration of functions by introducing new combinations of existing folds. This powerful mechanism was key to some significant evolutionary innovations, such as multicellularity and the vertebrate immune system. A method for reconstructing this important evolutionary process is urgently needed.

Results: Here, we introduce a novel, event-based framework for studying multidomain evolution by reconciling a domain tree with a gene tree, with additional information provided by the species tree. In the context of this framework, we present the first reconciliation algorithms to infer domain shuffling events, while addressing the challenges inherent in the inference of evolution across three levels of organization.

Conclusions: We apply these methods to the evolution of domains in the Membrane associated Guanylate Kinase family. These case studies reveal a more vivid and detailed evolutionary history than previously provided. Our algorithms have been implemented in software, freely available at http://www.cs.cmu.edu/~durand/Notung.

Background

Reconstruction of the history of change in a protein family provides valuable insight into processes of mutation and selection. Evolutionary reconstruction can reveal the context and order in which changes occurred, distinguish between shared history and convergent evolution, and identify interacting mutations that together result in a functional shift. Further, considering protein evolution in the context of species evolution makes it possible to correlate mutations with metabolic, physiological, and morphological changes, indicating which mutations are likely to be functionally important.

Despite tremendous advances in molecular evolution and phylogenetics, methods for reconstructing the evolutionary history of *multidomain* families are lacking. Genes that encode this large and important class of proteins are characterized by a mosaic of sequence fragments that encode structural or functional modules, called *domains*. Multidomain families are central to the two-component histidine kinase signaling systems that are the backbone of cellular communication in prokaryotes. In metazoans, the expansion of multidomain families drove the evolution of cell signaling and cell adhesion. In human health, multidomain families are implicated in tissue repair, apoptosis, inflammation response, antigen recognition, and innate immunity.

Multidomain families evolve via *domain shuffling* (Figure 1), a process that includes insertion, internal duplication, and deletion of domains. Because gain, loss, or replacement of a domain that encodes specificity can result in an immediate and dramatic change in function, domain shuffling enables rapid evolution

* Correspondence: durand@cmu.edu
[1]Department of Biological Sciences, Carnegie Mellon University, Forbes Ave, Pittsburgh, PA, 15213, USA
Full list of author information is available at the end of the article

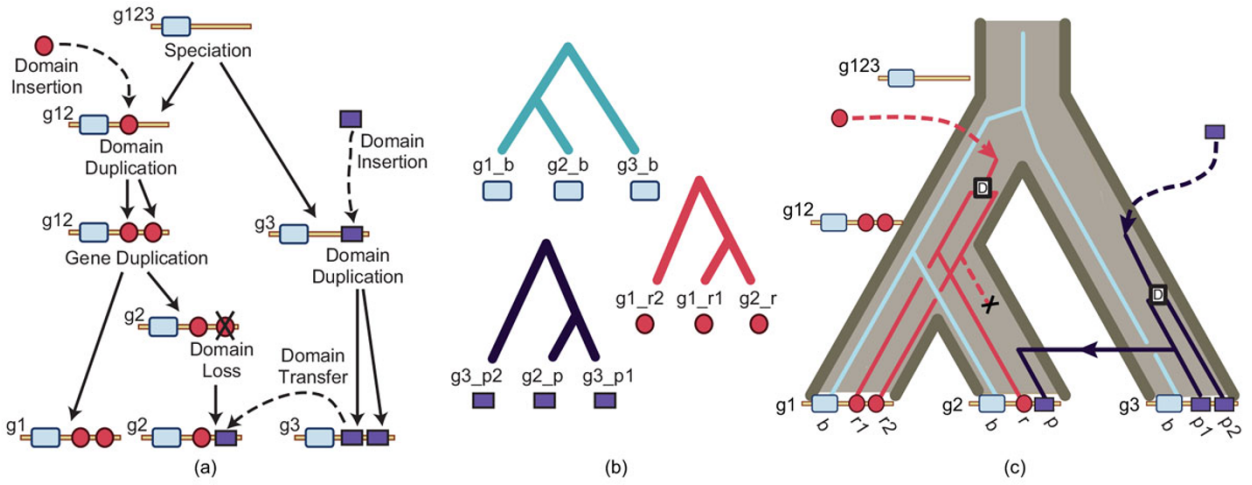

Figure 1 Schematic of multidomain evolution. (a) A hypothetical multidomain family evolving by gene duplication and domain shuffling. **(b)** Trees representing the history for each domain in the gene family. **(c)** The evolutionary history of the same family showing the domains evolving in the gene tree. Reconciliation correctly infers 2 domain duplications, 1 domain transfer and 1 domain loss. The gene family (locus) tree is shown in brown; black squares represent domain duplications.

of functional variation within gene families that perform core molecular functions.

Understanding the evolution of multidomain protein function requires understanding how domain architectures change over time. Simple domain architecture (DA) models have been used to achieve the computational efficiency necessary for genome-scale analyses [1,,3,4], and work cited therein]. In the DA model, a multidomain sequence is treated as a set or sequence of tokens (e.g. domain names or domain database identifiers [5-10]) representing its domain composition. The DA model has been used to study domain co-occurrence and variation in the domain repertoire across taxonomic lineages [11-13], plasticity in domain order [14,15], domain occurrence graphs [16,17], and domain promiscuity, i.e., the propensity of a domain to co-occur with many other domains [3,18-20].

In a phylogenetic context, patterns of domain gain and loss have been investigated by treating domain architectures as binary character data; a domain is either present or absent in a given architecture. Given a tree with architectures on the leaves, Wagner and Dollo parsimony have been used to infer ancestral domain architectures and the history of domain gains and losses [2,21-24]. Inferring multidomain trees by applying standard phylogenetic methods to domain architectures treated as character data has been proposed, either by calculating the pairwise edit distance between architectures [25] or by employing a parsimony model [21,26]. However, these approaches have not been applied in practical settings. Approaches to infer ancestral states using domain phylogenies have also been proposed [27-29], but these models have resulted in NP-complete optimization problems.

The benefits of the DA model include computational and conceptual tractability. However, it relies on several unrealistic simplifying assumptions: First, the DA model ignores sequence variation within domain superfamilies, treating all instances of a domain family as indistinguishable. Second, the DA model captures change in the form of domain gain and loss, but is not sufficiently powerful to infer the events that caused the change. For example, a domain gain could result from a domain insertion or an ancestral duplication followed by losses. Without an explicit event model, it is not possible to distinguish between these two scenarios. Third, the DA model will make incorrect inferences and underestimate the degree of domain shuffling activity in the presence of parallel gains or losses, as illustrated in Figure 2.

Here, we introduce a reconciliation-based framework for multidomain event inference. Reconciliation is the process of inferring evolutionary events by comparing the phylogenies of entities at two levels of biological organization. Given a rooted, binary gene tree, a rooted, binary species tree, and a mapping from extant genes to extant species, reconciliation seeks to infer the association between ancestral genes and ancestral species and the optimal history of gene duplications, gene losses, and horizontal gene transfers (HGTs) that explains this association.

Reconciliation for the duplication-loss (*DL*) model was first proposed by Goodman *et al.* [30] and formalized by Page [31] for a parsimony model. Hallett and Lagergren [32] introduced models with transfers and proved that

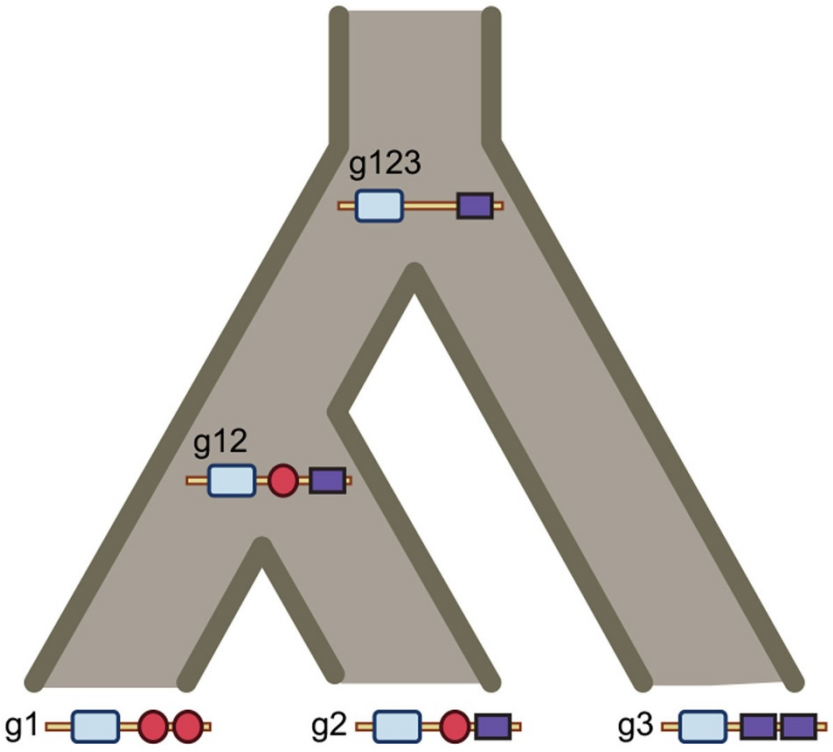

Figure 2 Domain architecture gain/loss model. Ancestral domain architectures for the hypothetical family in Fig. 1. Wagner parsimony applied to the the DA model infers 3 gains and 1 loss, underestimating the true events in the "known" history. The ancestral domain architectures inferred with Wagner parsimony are also incorrect.

reconciliation under the duplication-transfer (DT) model is NP-complete. The field has expanded with further algorithmic development for both the DL and DT models [33-40], and work cited therein]. Transfers introduce complications that do not occur in a duplication-only model. Transfers introduce degeneracy: there may be more than one optimal combination of transfers, duplications, and losses that gives rise to the same pattern of tree incongruence. Some reconciliation programs generate a single solution, selected at random. More recently, programs that generate all solutions have become available [38,41]. Transfers also introduce temporal constraints: a transfer can only occur between contemporaneous taxa. An event history is only biologically valid if it is *temporally feasible*; that is, if there exists a partial ordering of the nodes in the species tree that satisfies the temporal constraints imposed by the transfers in the inferred event history. There is no known constructive algorithm for generating minimum cost, temporally feasible DTL-reconciliations. Instead, dynamic programming is used to construct candidate minimum-cost reconciliations, which are then tested for temporal feasibility. A restricted model that only considers transfers between contemporaneous species, reviewed in [35,40], avoids this problem, but requires a tree with branch lengths in units of time, which are frequently difficult to estimate.

Here we present an event inference algorithm for reconstructing domain evolution. We define a set of domain shuffling events that includes domain duplication, domain loss, domain insertion within the same genome, and horizontal domain transfer between different genomes. We also consider a restricted model that allows domain insertion, but not horizontal transfer of genes or domains. In the context of this model, the events in the history of a multidomain family are inferred by reconciling a domain tree with a gene tree that has been previously reconciled with a species tree. This procedure also yields the timing of those events relative to gene and species divergences, as well as ancestral domain content. Consideration of the co-evolution of domains with both genes and species enables our algorithm to distinguish between domain losses and gene losses, and between domain insertions within the same genome and domain transfers across genomes. Further, our algorithm can determine whether a domain tree co-divergence is due to a species divergence (i.e., a speciation) or a gene divergence (i.e., gene duplication or transfer). In order to ensure the biological validity of the inferred evolutionary histories, we present criteria for temporal feasibility that

capture the complex temporal constraints associated with reconciliation involving domains, genes, and species. The implementation of these algorithms is freely available at http://www.cs.cmu.edu/~durand/Notung. To illustrate the inferential power of our approach, we apply our methods to the Membrane associated guanylate kinase (Maguk) family.

Results and discussion

Model

Given a rooted, binary gene tree, $T_{GS} = (V_G, E_G)$, a rooted, binary domain tree, $T_D = (V_D, E_D)$, and a mapping, M_L^{DG} from $L(T_D)$, the leaves in the domain tree, to $L(T_{GS})$, the leaves in gene tree, the goal of *domain shuffling event inference* is to infer the association between ancestral taxa in the species, gene, and domain trees and the set of events that best explains this association. In our model, we define the following set of events:

Co-divergence (C) A bifurcation in the domain tree that arose through a bifurcation in the gene tree. The gene tree bifurcation may have arisen via speciation (C_S), gene duplication (C_D), or gene transfer (C_T).

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

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

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

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

Reconciliation is the process of inferring $M^{DG} : V_D \rightarrow V_G$, the association between ancestral domains and ancestral genes. The result is a *reconciled domain tree,* $T_{DG} = (V_D, E_D)$, in which every node, $d \in V_D$, is annotated with $M^{DG}(d) = g$, where g is the ancestral gene that contained domain d; $\mathcal{L}$ (d) , the domains lost on the edge leading to d; and E(d), the event that caused the divergence at d. Co-divergences and domain duplications correspond to internal nodes in T_{DG}. Each insertion and transfer corresponds to an edge, $(d_1, d_2) \in E_D$, where d_2 is the inserted (or transferred) domain and d_1 is the donor domain. In a parsimony framework, the cost of a reconciliation is $\kappa = \sum_{\forall \varepsilon} \kappa_\varepsilon \cdot n_\varepsilon$, where κ_ε is the cost of event ε and n_ε is the number of occurrences of ε in the reconciliation.

Formally, we define the problem of inferring a domain event history as follows:

Domain Event Inference with Transfers (DE-DTL)

Domain events: $\{C_S, C_D, C_T, \mathcal{D}, \mathcal{T}, I, \mathcal{L}\}$.

Input: A rooted, binary domain tree, $T_D = (V_D, E_D)$; a rooted, binary, *DTL*-reconciled gene tree, T_{GS}; and a leaf mapping, $M_L^{DG} : L(T_D) \rightarrow L(T_{GS})$.

Output: The set of all temporally feasible, domain shuffling histories T_{DG} that minimize

$$\kappa = \kappa_{\mathcal{D}} n_{\mathcal{D}} + \kappa_{\mathcal{T}} n_{\mathcal{T}} + \kappa_I n_I + \kappa_{\mathcal{L}} n_{\mathcal{L}}.$$

Solving DE-DTL entails challenges that do not arise in gene tree-species tree reconciliation. First, when a domain co-divergence is inferred, additional information is required to determine whether the co-divergence is the result of a speciation, a gene duplication, or a gene transfer. Second, domain insertions are horizontal events between genes in the same species. In contrast, domain transfers are horizontal events between different species. An extra test is needed to ensure that the correct event is inferred. Third, a missing taxon in the domain tree may be due to a domain loss or the loss of a gene. If the latter, then the loss should not be included in the event cost of the domain-gene reconciliation. Finally, when testing for temporal feasibility, temporal constraints arising from gene transfers, domain transfers, and domain insertions must all be considered.

Notation: Given a tree $T_i = (V_i, E_i)$, ρ_i designates the root of the tree. For nodes $u, v \in V_i$, $p(v)$ denotes the parent of v and $l(v)$ and $r(v)$ denote the left child and right child of v, respectively. If u is on the path from v to ρ_i, then u is an ancestor of v, designated $u \geq_i v$, and v is a descendant of u, designated $v \leq_i u$. If $v \not\geq_i u$ and $u \not\geq_i v$, then u and v are *incomparable*, designated $u \lesseqgtr_i v$. Given a reconciled gene tree T_{GS} and a reconciled domain tree T_{DG}, $s = M^{GS}(g)$ is the species $s \in V_S$ that contained gene $g \in V_G$ and $g = M^{DG}(d)$ is the gene g that contained domain $d \in V_D$. The species containing d is $s = M^{DS}(d)$, where $M^{DS}(d) = M^{GS}(M^{DG}(d))$.

Domain shuffling with transfers and insertions

Here, we present an algorithm for the domain event inference problem for multidomain families evolving according to a *locus model* [42], in which novel domain arrangements arise through internal duplication, loss, and insertion of domains into an existing gene. This restriction justifies the premises that the history of the family as a whole can be described by a tree. This assumption is consistent with the existence of promiscuous domains that lend themselves to insertion in new chromosomal environments [1,3,43,44] and reports of young genes that arose through duplication of existing genes, followed by acquisition of additional domains [2,45-48]. Moveover, domain insertion into an existing gene is more likely to be viable since all regulatory and termination signals required for successful transcription are present. In addition, we assume that domain insertions and transfers only involve domains within the same gene family. In other words, for a given domain family, we assume that the domain instances that appear in the

gene family under consideration form a clade in the domain tree.

In the context of this model, we introduce an algorithm (Alg. 1) for inferring domain events by reconciling a domain tree with a gene tree that has been previously reconciled with a species tree.

Algorithm 1 DE-DTL

Input: T_S; T_{GS}; T_D; $M_L^{DG} : L(T_D) \rightarrow L(T_G)$

Output: $\left\{T_{DG}^1 \ldots T_{DG}^f\right\}$, κ

1 $T_{GS}^* = addLoss\ (T_{GS},\ T_S)$

2 $costCalc\ (\rho_D,\ T_{GS}^*,\ T_S)$

3 $\left\{T_{DG}^1 \ldots T_{DG}^m\right\} = traceback\ (\rho_D,\ T_{GS}^*,\ T_S)$

4 $\left\{T_{DG}^1 \ldots T_{DG}^f\right\} = checkFeasibility\ \left(\left\{T_{DG}^1 \ldots T_{DG}^m\right\}\right)$

The algorithm proceeds in four steps. First, an additional data structure, T_{GS}^*, is constructed. T_{GS}^* consists of an extended reconciled gene tree that contains additional nodes and leaves representing taxa that are missing due to gene losses. This additional data structure is used to determine whether or not the donor and recipient of a horizontal event are in the same genome or a different genome, and to distinguish between domain losses and gene losses. Next, candidate reconciliations are generated in two passes over the domain tree. In the first pass, the dynamic program costCalc visits each $d \in V_D$ in post-order and determines the cost of the subtree rooted at d for each possible event at d. This information is stored in the cost and event tables, K_d and H_d. In the second pass, traceback traverses T_D top-down to generate candidate reconciliations from the information stored in the cost and event tables. In general, there may be more than one optimal set of domain events that reconcile T_D with T_{GS}. The second pass generates all candidate event histories of minimum cost, $T_{DG}^1 \ldots T_{DG}^m$, where m is the number of candidate histories. In the final step, each candidate history is tested for conflicting temporal constraints. The output is the set of all temporally feasible histories, $\left\{T_{DG}^1 \ldots T_{DG}^f\right\}$, where f is the number of feasible, optimal reconciliations.

Our domain shuffling event inference algorithm is based on the existing framework for gene-species tree reconciliation with a DTL model [36,49], but contains additional features to address the complications that arise in reconciliation with three nested trees. We discuss these features in detail, here.

`addLoss`: We construct $T_{GS}^* = \left(V_G^*,\ E_G^*\right)$ from T_{GS} by placing pseudonodes on each edge $(p(g), g) \in E_G$, on which losses occurred. Each pseudonode represents an ancestral gene that was present in the species lineage from $M^{GS}(p(g))$ to $M^{GS}(g)$, but cannot be observed because of a gene loss in one child of each of the intervening species. Let $s_1 \cdots s_l$ be the species between $M^{GS}(g)$ and $M^{GS}(p(g))$ that are absent from T_G due to gene losses. We insert pseudonodes $\varphi_{s1} \ldots \varphi_{sl}$ between g and $p(g)$ such that φ_{s1}becomes the new parent of g, φ_{si} is the parent of $\varphi_{si-1}$for $i = 2 \ldots l$, and $p(g)$ becomes the new parent of φ_{sl}. For each pseudonode, φ_s, we attached a pseudoleaf, λ_s, where $s\prime = l(s)$, if $M^{GS}(g) \lesssim l(s)$, and $s\prime = r(s)$, otherwise. In other words, S' is the child of s that is not on the path from $M^{GS}(g)$ to $M^{GS}(p(g))$ and $(s\prime, s)$ is the species tree branch on which the loss occurred. Note that a pseudoleaf in T_{GS}^* may correspond to an internal node in T_S, because if a gene is missing from an entire clade of species, then the most parsimonious explanation is a single gene loss in the root of the clade.

The addition of pseudonodes allow for more precise estimation of the association between a node in the domain tree and a lineage in the gene tree. Consider, for example, the evolutionary history of the hypothetical family shown in Figure 3, and the reconcilied gene and domain trees for this family, shown in Figure 4. We can infer that the ancestral domain associated with node u in T_D was in a genome on the Eudicot-Rosid lineage because of the position of the pseudonode φ_R, between u and d1_*g*1_*A*. Without the pseudonode, it would not be possible to determine whether or not u predates the divergence between Apple and Berry.

Pseudoleaves are also used to distinguish between gene and domain losses. The domain tree in Figure 4(c) has three stubs indicating missing taxa. Two of these are associated with gene losses (ℓ_B and ℓ_C), indicating that only one of the missing domains is due to a domain loss (*d*2_*g*2_*B*).

`costCalc`: Once the extended reconciled tree has been constructed, the first pass takes T_D and T_{GS}^* as input and calls costCalc(d) for each $d \in V_D$ in post-order (Alg. 2). The gene association and event at d depend on g_l and g_r, the genes associated with the children of d. The outer loop in cost Calc enumerates all possible (g_l, g_r) pairs, and for each pair determines the gene associations and events implied by $M^{DG}(l(d)) = g_l$ and $M^{DG}(r(d)) = g_r$. The cost of each configuration is stored in K_d. A tuple consisting of the event and the mappings of the children of d are stored in H_d. The logic for these assignments is as follows.

Domain duplication is the only event that results in children mapped to comparable genes. Thus, if g_l and g_r are comparable, then $\varepsilon = \mathcal{D}$ and the gene associated with d is $g = lca(g_l, g_r)$. Horizontal events and co-divergences both result in $g_l \lesssim g_r$. Therefore, if g_l and g_r are incomparable, both possibilities are considered. For the co-divergence case, the associated gene is again $g = lca$

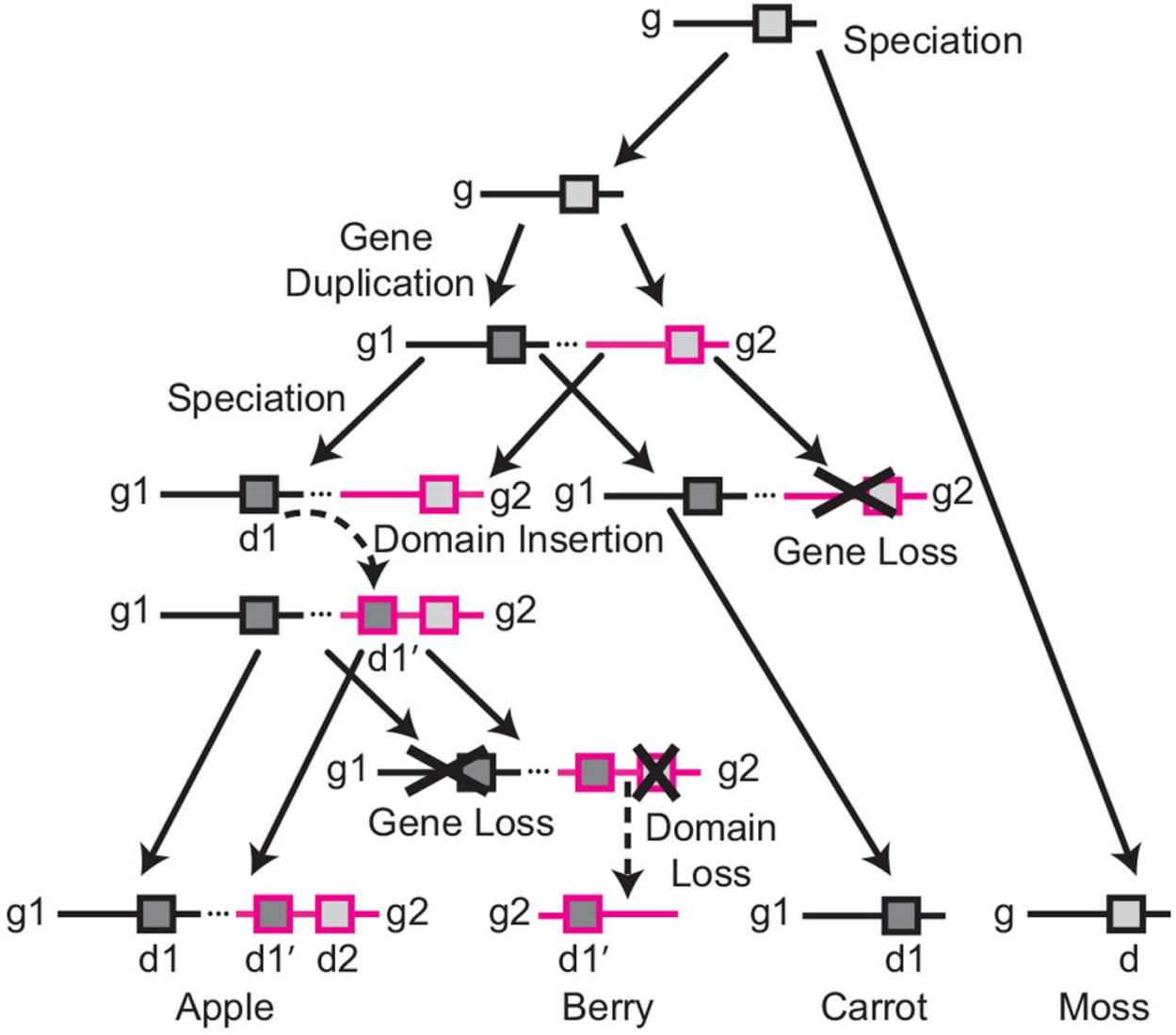

Figure 3 Evolution of a hypothetical multidomain gene family. Domain instances are represented by grey squares. Duplicated genes in the same genome are connected by dotted lines.

(g_l, g_r). The type of co-divergence is determined by inspecting the gene event associated with gene node g in T^*_{GS}. For the horizontal case, the event is a domain insertion if the donor and recipient gene are in the same genome and a domain transfer if they are in different genomes. For both insertions and transfers, either g_l or g_r may be associated with d; i.e., either g_l or g_r could be the donor of the domain.

For each scenario, the cost of domain losses must also be determined, excluding cases where domain loss is due to gene loss. Recall that pseudonodes correspond to gene losses. Therefore, the number of losses, $n_{\mathcal{L}}$, on the edge from d to $p(d)$ is the number of non-pseudonodes between $g = M^{DG}(d)$ and $g_p = M^{DG}(p(d))$. If g and g_p are not pseudonodes, then this quantity is $\Delta(g) - \Delta(g_p) - 1$, where $\Delta(g)$ is the depth of g in the original gene tree, T_G. However, if g is a pseudonode, φ, then $\Delta(g)$ is undefined. We define $\Delta(\varphi)$ to be the depth of the first non-pseudonode ancestor of φ in T^*_{GS}. In other words, $\Delta(\varphi) = \Delta(u)$, where $u \geq_G \varphi$ and there exists no $v \in V_G$, such that $u \geq_G v \geq_G \varphi$. This effectively jumps over all pseudonodes between φ and u. If g is a pseudonode and its first non-pseudonode ancestor is a loss, then directly using the depth of the first non-pseudonode ancestor will fail to count this loss. In this case, the number of losses is incremented by one (lines 15 and 16). No such correction is needed when g_p is a pseudonode.

This formulation allows for efficient calculation of the number of losses under each scenario considered in costCalc. If E(d) is a co-divergence, then the number of

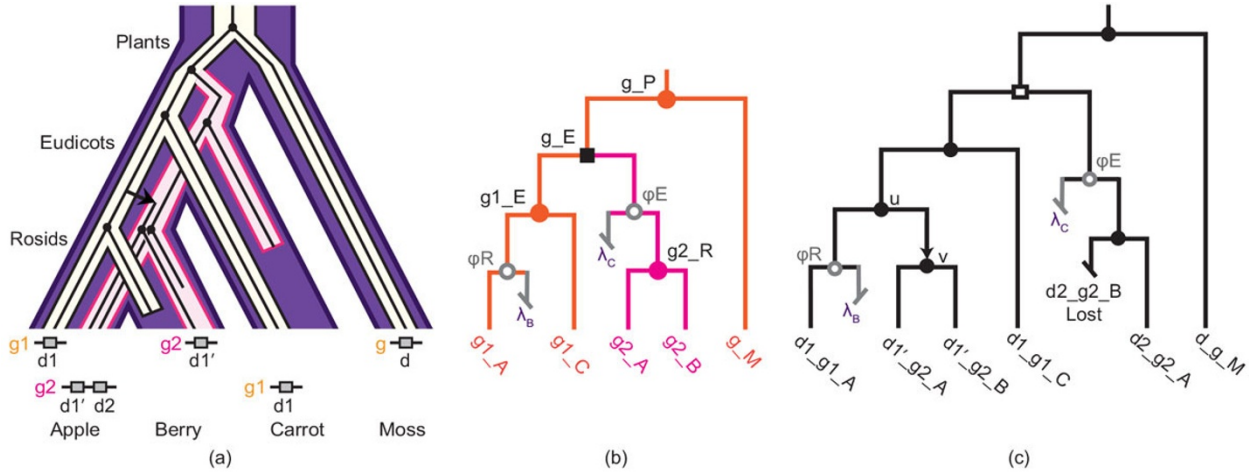

Figure 4 Domain insertions in the presence of gene loss. **(a)** Embedded trees showing the co-evolution of domains, genes, and species in the hypothetical family in Fig. 3. **(b)** An extended reconciled gene tree for the gene family. Inferred gene losses (ℓ_B and ℓ_C) and pseudonodes (open circles, φ_R and φ_E) representing the location of the missing taxa in the gene tree are shown in grey. The pseudonodes are used to distinguish between gene losses and domain losses in the reconciled domain tree. Gene duplication is represented as a black square; filled circles represent co-divergences. **(c)** A reconciled tree for the domain family, showing a domain insertion (arrow, edge (*u, v*)) and a domain loss (*d2 g2 B*). Domains that are missing due to gene loss (ℓ_B and ℓ_C) are shown in grey. Co-divergence due to gene duplication is represented by an open square.

losses is $(\Delta(g_l) - \Delta(g) - 1) + (\Delta(g_r) - \Delta(g) - 1)$. If E ($d$) is a duplication, g_l, g_r, and g should all be at the same depth, and the number of losses is $(\Delta(g_l) - \Delta(g)) + (\Delta(g_r) - \Delta(g))$.

`traceback`: Once the first pass is complete, the cost and event tables are filled for each node in T_D. The traceback algorithm constructs a minimum-cost reconciliation using these tables in a pre-order traversal. At every node $d \in V_D$, the appropriate tuple in the event table H_d is used to assign an event to E(d) and determine the labels of the genes associated with the children of d. The losses that occurred between d and $p(d)$ are inferred in a climbing procedure [38,50]. For each ancestral node g that is missing between $M^{DG}(d)$ and $M^{DG}(p(d))$, a loss is inferred in g', the child of g that is incomparable to $M^{DG}(d)$. If g is a pseudonode in T_{GS}, then g' is a gene loss, λ_s. Otherwise, a domain loss in g' is inferred.

Algorithm 2 DE-DTL: CostCalc

Input: T^*_{GS}; T_D; $M^{DG}_L : L(T_D) \rightarrow L(T_G)$

Output: K_d, H_d $\forall d \in V_D$; ν

costCalc(d) {
1 ***if*** $d \in L(T_D)$ {
2 ***for each*** $g \in V^*_G$ {
3 ***if*** $M^{DG}_L(d) \lesssim g$ {*table*(d, g, ∞, ∞, *null*, *null*) }
4 ***else*** {
5 $n_{\mathcal{L}} = \Delta(M^{DG}_L(d)) - \Delta(g)$
6 *table*(d, g, C_S, $n_{\mathcal{L}}$, *null*, *null*)
7 }
8 }
9 *return*
10 }
11 *costCalc*($l(d)$), *costCalc*($r(d)$)
12 ***for each*** $(g_l, g_r) \in V^*_G \times V^*_G$ {
13 $g = lca(g_l, g_r)$
14 $n_{\mathcal{L}} = \Delta(g_l) + \Delta(g_r) - 2 \cdot (\Delta(g) + 1)$
15 ***if*** (g_l is pseudonode) { $n_{\mathcal{L}}$ + + }
16 ***if*** (g_r is pseudonode) { $n_{\mathcal{L}}$ + + }
17 ***if NOT*** $(g_l \lesssim g_r)$ {
18 *// Duplication case;*
19 $\varepsilon \leftarrow \mathcal{D}$
20 $n_{\mathcal{L}} = n_{\mathcal{L}} + 2$
21 *table*(d, g, ε, $n_{\mathcal{L}}$, g_l, g_r)
22 }
23 ***else*** {
24 *// Co-divergence case;*
25 ***if*** $E(g) = \mathcal{D}$ {$\varepsilon \leftarrow C_{\mathcal{D}}$} *// Duplication*
26 ***else if*** $E(g) = \mathcal{T}$ {$\varepsilon \leftarrow C_{\mathcal{T}}$} *// Transfer*
27 ***else*** { $\varepsilon \leftarrow C_S$ } *// Speciation*
28 *table*(d, g, ε, $n_{\mathcal{L}}$, g_l, g_r)
29 *// Domain insertion or transfer case;*
30 ***if*** $M^{GS}(g_l) = M^{GS}(g_r)$ { *// Same genome*
31 *// Domain insertion case;*
32 *table*(d, g_l, I, 0, g_l, g_r) *// g_l to g_r*
33 *table*(d, g_r, I, 0, g_l, g_r) *// g_r to g_l*
34 } ***else if*** $M^{GS}(g_l) \lesssim_S M^{GS} g_r$ {
35 *// Domain transfer case;*
36 *table*(d, g_l, $\mathcal{T}$, 0, g_l, g_r) *// g_l to g_r*
37 *table*(d, g_r, $\mathcal{T}$, 0, g_l, g_r) *// g_r to g_l*
38 }

```
39   }
40 }
table(d, g, ε, n_L , g_l, g_r ) {
41  κ ← κ(ε) + K_l(d) [g_l] + K_r(d) [g_r] + κ_L · n_L
42 if κ < K_d[g] {
43      K_d[g] ← κ
44      H_d [g] ← list ((ε, g_l, g_r))
45 } else if κ = K_d [g] { list (H_d [g], (ε, g_l, g_r )) }
46 }
```

If there is more than one optimal candidate reconciliation, T_D is traversed repeatedly until all minimum cost histories have been generated. The traceback requires no modification to address reconciliation on three levels and is essentially the same as the traceback algorithm for gene tree-species tree reconciliation, as described in [38].

`checkFeasibility`: Once all candidate reconciliations have been returned, their temporal feasibility is checked to ensure that each solution is valid. This requires criteria for temporal feasibility that accommodate the interplay of domain evolution with gene and species evolution. To be temporally feasible, an inferred domain event history must satisfy two criteria. First, it must be possible to assign a timestamp to every species tree node, such that the timestamps are consistent with the temporal constraints imposed by the combined set of gene and domain transfers. Second, it must be possible to assign a timestamp to every gene tree node, such that the timestamps are consistent with the temporal constraints imposed by the combined set of domain transfers and domain insertions.

For the first criterion, we introduce a *species timing graph*, $G_t^S = (V_t^S, E_t^S)$, in which the nodes represent species (i.e., $V_t^S = V_S$) and the edges represent temporal constraints introduced by gene and domain transfers, as well as the temporal relationships imposed by directed tree edges. If gene g is horizontally transferred from species s_1 to s_2, then every domain in g is also implicitly transferred from s_1 to s_2. Thus, every gene transfer, (g_1, g_2), corresponds to one or more edges in the domain tree. Let $\Lambda_G \subset E_G^*$ and $\Lambda_D \subset E_D$ be gene and domain transfer edges in T_{GS}^* and T_{DG}, respectively, and let Λ_G^- be the set of edges (d_1, d_2), such that $\exists (g_1, g_2) \in \Lambda_G$, where $g_1 = M^{DG}(d_1)$, $g_2 = M^{DG}(d_2)$. Then, $\Lambda_G^{-1} \cup \Lambda_D$ represents all the temporal constraints arising from horizontal transfers of both genes and domains.

The temporal constraints on species are encoded in G_t^S by adding edges to E_t^S that represent the following three temporal constraints:

1 If species $s_i = p(s_j)$, then s_i must have predated s_j. $\forall (s_i, s_j) \in E_S$, add (s_i, s_j) to E_t^S.

2 If a transfer from species s_i to species s_j occurred, then s_i and s_j must have co-existed. Further, the parent of s_j must have predated s_i, and vice versa. $\forall (d_i, d_j) \in \Lambda_G^{-1} \cup \Lambda_D$, add $(p(s_i), s_j)$ and $(p(s_j), s_i)$ to E_t^S, where $s_i = M^{DS}(d_i)$ and $s_j = M^{DS}(d_j)$.

3 If two transfers occur in the same lineage in the reconciled domain tree (i.e., one is ancestral to the other), then the species corresponding to the donor and recipient of the more ancestral event must have occurred no later than the donor and recipient species of the more recent event. Let (d_1, d_2) and (d_1', d_2') be a pair of transfers in $\Lambda_G^{-1} \cup \Lambda_D$, such that $d_2 \geq_D d_1'$. Then $s_1 = M^{DS}(d_1)$ and $s_2 = M^{DS}(d_2)$ must have occurred no later than species $s_1' = M^{DS}(d_1')$ and $s_2' = M^{DS}(d_2')$. Add $(p(s_1), s_1')$, $(p(s_1), s_2')$, $(p(s_2), s_1')$, and $(p(s_2), s_2')$ to E_t^S.

Note that the third constraint considers all pairs of comparable transfers. When mapping gene transfer events to the reconciled domain tree, each pair of transfers that is comparable in T_G corresponds to at least one pair of edges in Λ_G^- that is comparable in T_D.

For the second criterion, we construct a *gene timing graph*, $G_t^G = (V_t^G, E_t^G)$, in which the nodes represent genes and the edges represent temporal constraints implied by domain transfers, Λ_D, and insertions, $\circ_D$, where $\circ_D \subset E_D$ denotes the set of domain insertions in T_{DG}. The nodes in G_t^G are the nodes in T_{GS}^*, including pseudonodes (i.e., $V_t^G = V_G^*$). Edges are added to E_t^G to represent the following three temporal constraints:

1 If gene $g_i = p(g_j)$, then g_i must have predated g_j. $\forall (g_i, g_j) \in E_G^*$, add (g_i, g_j) to E_t^G.

2 If a domain was transferred or inserted from gene g_i to g_j, then g_i and g_j must have co-existed. Further, the parent of g_i must have predated g_j and vice versa. $\forall (g_i, g_j) \in \{(M^{DG}(d_1), M^{DG}(d_2)) \mid (d_1, d_2) \in \Lambda_D \cup \circ_D\}$, add $(p(g_i), g_j)$ and $(p(g_j), g_i)$ to E_t^G.

3 Let (d_1, d_2) and (d_1', d_2') be domain insertions and/or transfers in $\Lambda_D \cup \circ_D$, such that $d_2 \geq_D d_1'$. Then $g_1 = M^{DG}(d_1)$ and $g_2 = M^{DG}(d_2)$ must have existed no later than $g_1' = M^{DG}(d_1')$ and $g_2' = M^{DG}(d_2')$. Add $(p(g_1), g_1')$, $(p(g_1), g_2')$, $(p(g_2), g_1')$, and $(p(g_2), g_2')$ to E_t^G.

A domain event history is temporally feasible *iff* both timing graphs are acyclic. If either graph contains a cycle, then the candidate history is infeasible and is not reported. A modified topological sorting algorithm in $\Theta(|V_t| + |E_t|)$ [51] is used to test for cycles in G_t.

Complexity

Our algorithm infers domain shuffling events in polynomial time. addLoss constructs T^*_{GS} by adding pseudonodes to T_{GS}. The number of pseudonodes that can be added to the edge above a node in T_{GS} is at most h_S, where h_S is the height of T_S. Therefore, the complexity of addLoss is $O\left(|V^*_G|\right) = O\left(h_S|V_G|\right)$. costCalc visits each domain node $d \in V_D$ and loops through all pairs of gene nodes $\left(g_l, g_r\right) \in V^*_G \times V^*_G$. Since the calculations for each combination of d, g_l, and g_r requires only constant time, the total complexity is $O\left(|V_D||V^*_G|^2\right) = O\left(|V_D|h^2_S|V_G|^2\right)$.

traceback constructs a single candidate solution in a preorder traversal of the domain tree. Looking up the event, the number of losses, and mapping of each node requires constant time. For each node $d \in V_D$, the losses between d and $p(d)$ are calculated by climbing from $M^{DG}(d)$ to $M^{DG}(p(d))$. This requires $O\left(h^*_G\right)$ time for each d, where h^*_G is the height of T^*_{GS}. Therefore, the total complexity for returning a single condidate reconciliation in the second pass is $O\left(h^*_G|V_D|\right)$. Since the number of pseudonodes added to an edge in T_{GS} is bounded by h_S and there are at most h_G nodes that are not pseudonodes contributing to h^*_G, $h^*_G \leq h_S h_G$. As a result, the complexity of the second pass is $O(h_S h_G|V_D|)$.

To determine whether a candidate solution is temporally feasible, checkFeasibility tests both the gene and species timing graphs for cycles, using a topological sorting algorithm that runs in $O(|V_t| + |E_t|)$.

For the species timing graph, $V^S_t = V_S$. The number of edges in E^S_t depends on the three constraints described in the previous section:

1 $|E_S|$ edges are added to E^S_t, one for each species tree edge.

2 Two edges are added to E^S_t for each transfer in $\Lambda^-_G \cup \Lambda_D$. This results in the addition of at most $2|E_D|$ edges because $|\Lambda^-_G \cup \Lambda_D| < |E_D|$.

3 Four edges are added to E^S_t for every pair of comparable transfers in $\Lambda^-_G \cup \Lambda_D$. Since the number of pairs is bounded by $|E_D|^2$, the number of added edges is bounded by $4|E_D|^2$.

Combining all three constraints, the complexity of cycle checking in the species timing graph is $O(|V_S| + |E_S| + |E_D| + |E_D|^2)$. Because $|V_S| = O(|E_S|)$ and $|E_D| \geq |E_S|$, $|E_D|^2$ is the dominant term. Therefore, the complexity can be written as $O(|E_D|^2)$.

For the gene timing graph, $V^G_t = V^*_G$ and the number of edges in E^G_t depends on the three previously described constraints. The first constraint contributes E^G_t edges to E^G_t. In the worst case, the second and third constraints contribute the same number of edges for the gene timing graph as for the species timing graph, that is $O(|E_D|)$ and $O(|E_D|^2)$. Thus, the complexity of cycle checking in the gene timing graph is $O\left(|V^*_G| + |E^*_G| + |E_D| + |E_D|^2\right)$. Recall that $|E^*_G| = O\left(|V^*_G|\right) = O\left(h_S|V_G|\right)$. Because $|E_D| \geq |V_G|$ and $|E_D| \geq h_S$, this complexity can be written as $O\left(|E_D|^2\right)$.

Domain shuffling with insertions only

The *DE-DTL* model includes horizontal transfer of both genes and domains and is suitable for reconstructing the history of domain shuffling events in species that accept foreign DNA. However, this model is not appropriate for analysis of multidomain families in species that do not participate in genetic exchange with other species. For such families, we also consider domain shuffling inference for the restricted model without transfer events. This model is particularly well-suited to the large and complex multidomain families in vertebrates [12], in which HGT is thought not to occur.

Domain Event Inference without Transfers (DE-DL)

Domain events: $\{C_S, C_D, \mathcal{D}, I, \mathcal{L}\}$.

Input: A rooted, binary domain tree, $T_D = (V_D, E_D)$; a rooted, binary, *DL*-reconciled gene tree, T_{GS}; and a leaf mapping, $M^{DG}_L : L(T_D) \rightarrow L(T_G)$

Output: The set of all temporally feasible, domain shuffling histories T_{DG} that minimize

$$\kappa = k_{\mathcal{D}} n_{\mathcal{D}} + \kappa_I n_I + \kappa_{\mathcal{L}} n_{\mathcal{L}}.$$

The overall structure of the algorithm for DE-DL is the same as for DE-DTL, i.e., Alg. 1. However, since transfers are not allowed, the input gene tree must be reconciled with a species tree under the *DL* model, not the *DTL* model. An extended gene tree, T^*_{GS}, is constructed using the same procedure as before. With minor modifications, Alg 2 generates solutions to the DE-DL problem. In the co-divergence case, line 26 is omitted to exclude co-divergences with gene transfers $(C_{\mathcal{T}})$ from the event set. The domain transfer case (lines 34-37) is eliminated altogether. The second pass is identical for both problems, with the exception that transfers do not appear in the cost and event tables.

The worst-case time complexity for costCalc is the same as in DE-DTL, but because domain transfers are not considered and domain insertions are only allowed between genes in the same species, the number of mappings that can be considered is greatly restricted. This suggests a faster run-time in practice.

Determining temporal feasibility is a simpler procedure, since only the gene timing graph must be

constructed and tested for cycles. The complexity for gene tree timing graph is $O(|E_D|^2)$.

Case studies

Here we demonstrate the practical application of our approach using several examples from the Membrane-associated guanylate kinase (Maguk) family [52,53].

The Maguks are multidomain scaffolding proteins (Figure 5) that play important roles in cell-cell communication and adhesion including mediating cell polarity [54,55], cell proliferation [56], and synaptic plasticity [57,58]. Scaffolding proteins assemble the components of a signaling cascade in the appropriate configuration [59]. The multidomain architecture is integral to this function because each of the constituent domains is responsible for anchoring specific proteins to the signaling complex. Therefore, acquisition, loss, or replacement of a domain with another from the same family could result in an immediate and dramatic change in interaction partners.

All Maguks have an inactive Guanylate Kinase (*GuK*) domain, in combination with various adapter domains that anchor downstream pathway proteins to the scaffold. There are six Maguk subfamilies, each with a characteristic domain architecture (Figure 5). The identity, copy number, and order of the auxiliary domains is largely conserved within each subfamily, with minor variations.

To analyze the history of domain shuffling in the Maguk family, we require a species tree, a gene tree, and a tree for each domain of interest. For these case studies, we restrict our analysis to three species, human, mouse and chicken, for which the species tree is well known. For the gene tree, we require a phylogeny that represents the history of the entire gene family locus. This can be a challenge for multidomain families with variable domain content, including the Maguks, because it is not possible to obtain a full length sequence alignment, from which to construct a gene tree. In this analysis, we use the phylogeny of the *GuK* domain as a proxy for the history of the gene family. This domain is present in all Maguk architectures and is unique to the Maguk family. In Maguks, the *GuK* and *SH3* domains participate in intramolecular interactions that cause them to function together as a unit [56,60]. This suggests that the *GuK* domain is under tight structural and functional constraints and, hence, is unlikely to participate in domain shuffling, making it a suitable proxy for the history of the locus.

With this in mind, we constructed a *GuK* domain phylogeny (see Methods) for all members of the Maguk family in mouse, human, and chicken (Figure 6 and Fig. S1 in Additional file 1). Using this gene tree, we investigated possible domain shuffling for two Maguk constituent domains: the *L27* domain in the Membrane-associated Proteins, Palmitoylated (*MPP*) subfamily, and the *PDZ* domain, which is found in all Maguk subfamilies, except the calcium channel β (*CACNB*) proteins.

The L27 domain. The *MPPs* mediate protein complex formation at cell junctions and play a role in establishing cell polarity during development [54]. All *MPPs* contain a core *GuK-SH3-PDZ* domain architecture and, except *MPP1*, two N-terminal *L27* domains (Figure 5).

To determine whether the *L27* domains co-evolved by vertical descent with the Maguk structural core, we constructed trees from the sequences of the first and second *L27* domains (*L27-1* and *L27-2*) in human, mouse, and chicken *MPPs* (see Methods). Outside of the *MPP* subfamily, only a few proteins encoded in the human genome possess *L27* domains (*DLG1*, Lin7A/B/C, MPDZ, INADL). These proteins possess only a single *L27* domain, rather than a tandem pair. Structural studies partition single-copy and tandem *L27* domains into two distinct subtypes, with distant sequence homology [61]. This suggests that *MPP L27* domains are more closely related to each other than to other *L27* domains and, thus, satisfy the assumption of our algorithm, which only considers domains within the current gene family.

The *L27* phylogeny was reconciled with the *GuK* reference tree (Figure 7(a)) using the DE-DL model and event costs $\kappa_{\mathcal{D}} = 3$, $\kappa_I = 1.5$, and $\kappa_{\mathcal{L}} = 3$. The *L27-2* subtree (not shown) is consistent the hypothesis that the *L27-2* and *GuK* domains co-evolved without shuffling. However, the reconciled *L27-1* tree (Figure 7(b)) suggests that the *L27-1* domain in the *MPP2/6* ancestor was replaced by a copy of *L27-1* from the *MPP3/7* ancestor, by a single domain insertion (Figure 7(c)).

Phylogenetic error is of particular concern in domain tree reconstruction, because domain sequences tend to be short and weakly conserved [42]. To investigate the impact of phylogenetic error on the inferred *L27-1* domain insertion, we generated the 95% confidence set of trees with high likelihood scores [62,63], as described in Methods. Only two topologies are supported by at least 25% of the trees in the confidence set, as shown in the consensus network [64] in Figure 8. In both topologies, *MPP2/6/3/7* form a clade, which is consistent with the hypothesis that the *L27-1* domain was replaced in the *MPP2/6* ancestor.

Previous models of *MPP* evolution [52] predicted that the *L27-1* domains in *MPP2/6* and *CASK* share a unique, common ancestor. In contrast, our analysis predicts that the *L27-1* domains in *MPP2/6* and *MPP3/7* are more closely related. This has functional, as well as evolutionary, implications. A vertical descent model implies that *MPP2/6* are likely to be most functionally similar to *CASK*, their closest *MPP* relative. Our analysis

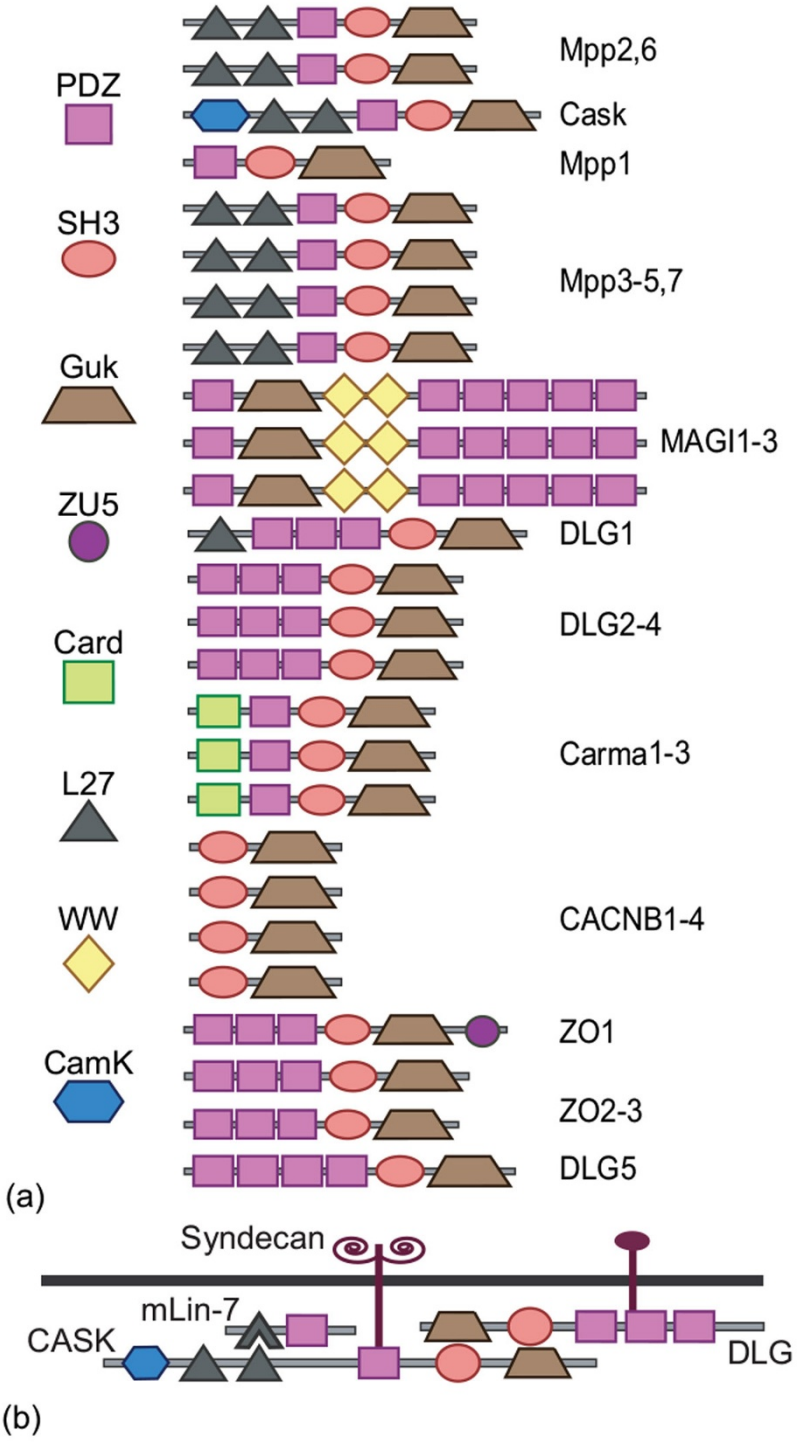

Figure 5 Multidomain Maguk gene family. **(a)** Domain architectures found in the Maguk family. Note that *DLG5* is a member of the *ZO* subfamily [53]. **(b)** Model of a scaffolding complex with two interacting Maguk proteins, adapted from [54].

predicts that *MPP2/6* are more likely to be similar to *MPP3/7*, with respect to *L27* mediated interactions. This is consistent with recent experimental results that show that *MPP2*, *MPP3*, and *MPP7* share a common *L27*-mediated complex formation mechanism, while *CASK* uses a different mechanism [65].

The PDZ domain. All members of the Maguk family, except for the *CACNBs*, have at least one *PDZ* domain.

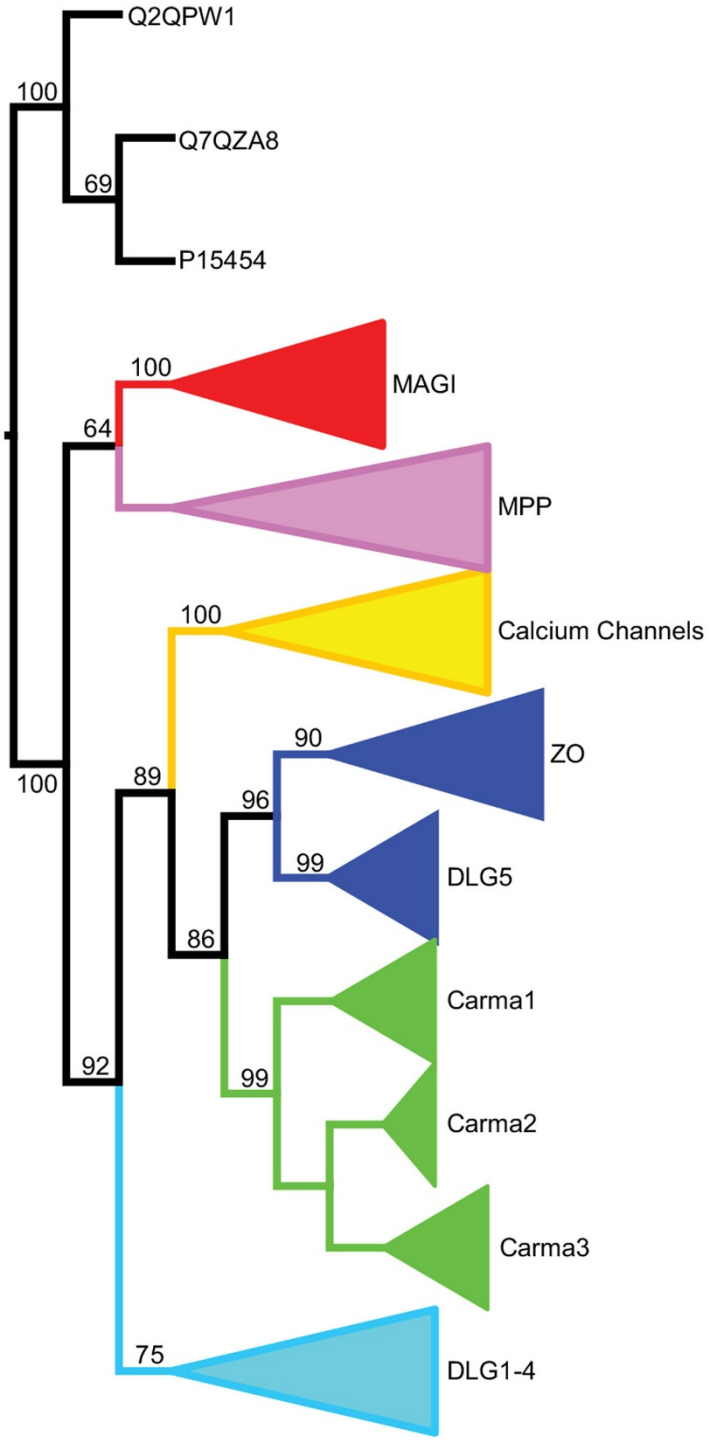

Figure 6 Phylogenetic relationships of Maguk subfamilies. Maximum likelihood phylogeny of *GuK* domain sequences. Clades containing paralogous genes from the same subfamily are collapsed. Edge weights are the number of bootstrap replicates, out of 100, supporting that edge.

Using our newly developed methods, we sought to understand how the varied number of *PDZ* domains arose in this family. Was there a single *PDZ* in the earliest Maguk that expanded independently in different subfamilies? Or was there a round of ancestral domain duplication followed reciprocal, independent losses?

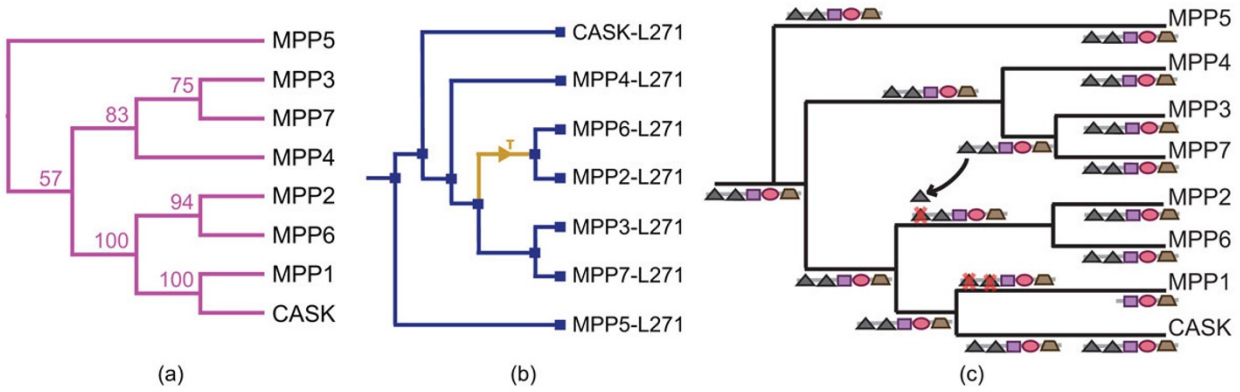

Figure 7 History of L27 domain shuffling. **(a)** Phylogeny of the *MPP* subfamily, based on the *GuK* domain maximum likelihood tree. **(b)** The *L27-1* domain phylogeny showing the inferred domain insertion event. **(c)** Reconstruction of the evolution of the *MPP* domain architecture, showing the replacement of the N-terminal *L27* domain in the common ancestor of *MPP2* and *MPP6*. Clades containing mouse, human, and chicken orthologs are collapsed.

The maximum likelihood (ML) tree for the *PDZ* domains (Fig. S2 in Additional file 1) suggests a number of domain shuffling events when reconciled with the *GuK* reference tree. However, many of the relationships in this tree are associated with low bootstrap values and may not reflect an accurate evolutionary history. To investigate the robustness of the inferred domain shuffling events, we generated the 95% confidence set of *PDZ* trees as described in Methods. All 121 trees in this set imply that a number of domain shuffling events occurred - some of these events are unique to a single tree, while others are common to a large fraction of the trees. We defined the support of an inferred event in this set to be the fraction of reconciled *PDZ* trees in which that event occurs (see Methods). The inferred events were scored and ranked based on this support score (Table S1 and Fig. S3 in Additional file 1). The reconciled trees from the *PDZ* confidence set were then ranked based on the number of high-scoring events found in their history. Tree 84 (Fig. S4 in Additional file 1) has nine of the top ten high-scoring insertion events. Only one tree has all top ten events, but that tree results in a much higher reconciliation cost (439.9), compared with a cost of 399.1 for Tree 84. We therefore use Tree 84 in the following analysis of *PDZ* domain evolution in the Maguks (Figure 9).

Our reconstructed history of *PDZ* domain shuffling shows several interesting trends that disagree with prior analyses based on Wagner parsimony. This is especially true for the *ZO* and *Carma* subfamilies.

All members of the *ZO/DLG5* subfamily have a cassette of at least three tandemly arranged *PDZ* domains (and a fourth in *DLG5*). It has been previously been proposed [52] that two domain duplications in the *ZO/DLG*

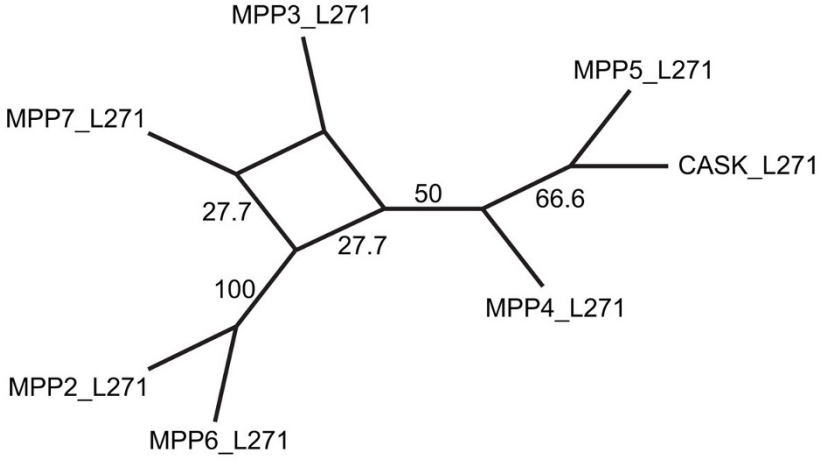

Figure 8 Consensus network for the L27-1 confidence set.

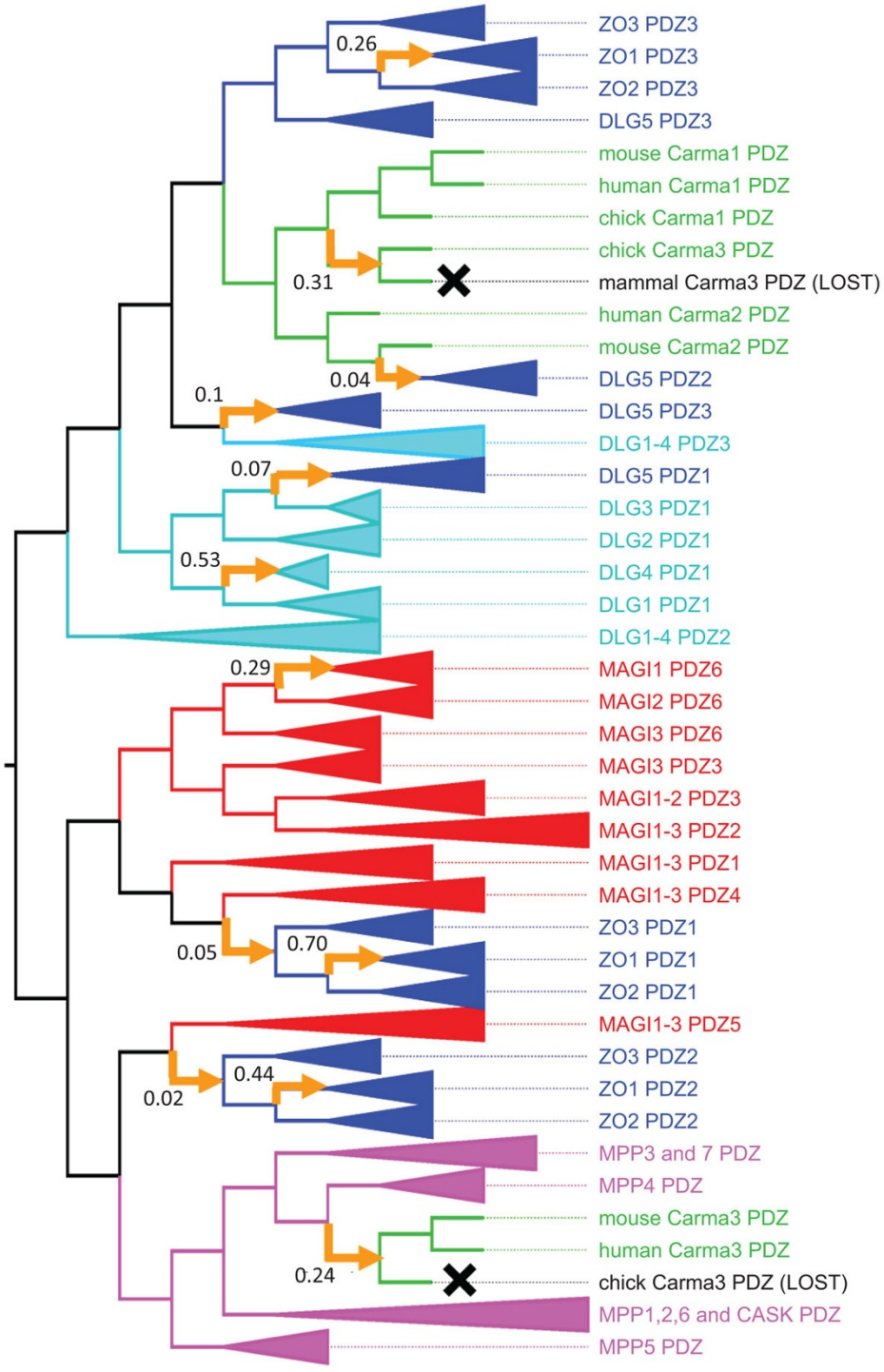

Figure 9 Maguk PDZ event history. Based on reconciled domain tree 84, shown in Fig. S2 (in Additional File 1). Leaf labels correspond to protein name, followed by domain name. Domains in each protein are numbered in N- to C-terminal order. Clades containing mouse, human, and chicken orthologs are collapsed, except in the *Carma* family, in which orthologous sequences are not monophyletic. Clades of paralogous genes from the same subfamily are also collapsed. All insertion events that are not within collapsed clades are shown (yellow arrows, annotated with event support values), including seven of the nine high-scoring events. Domain losses are indicated in gray. Gene losses not shown.

ancestor gave rise to this cassette, which was subsequently inherited by all *ZOs* and *DLG5*, with an additional duplication resulting in the fourth *PDZ* in *DLG5*.

Our reconstruction, based on domain tree reconciliation, tells a very different story (Figure 10). First, the *ZO* ancestor had a single *PDZ* domain that was vertically inherited in *ZO2* and *ZO3*, but lost in *ZO1*. Second, the cassette expansion was the result of a domain insertion from *MAGI* into the common ancestor of *ZO2* and *ZO3*. While these insertions are weakly supported, the fact that both the donor domains (*PDZ4* and *PDZ5* in *MAGI*) and the recipient domains (*PDZ1* and *PDZ2* in *ZO2* and *ZO3*) are adjacent is suggestive. Third, the *PDZ* cassette in *ZO1* was the result of three, highly supported domain insertion events from *ZO2* in amniotes. Considering the fact that these three *PDZs* are tandemly located in the domain architecture, it is possible that this was the result of a single insertion event involving all three *PDZ* domains simultaneously. Interestingly, the C-terminal *PDZ* domain in *DLG5* and the C-terminal *PDZ* in the *ZO* proteins were both vertically inherited from the same ancestral copy. The other *DLG5 PDZ* domains were the result of insertions from the *DLG* and *Carma* subfamilies.

Also of particular interest are the high-scoring insertion events in the *Carma* subfamily. All *Carma* genes have a single *PDZ* domain that is predicted to be the result of vertical inheritance in Wagner parsimony analysis. According to our domain tree reconciliation analysis, however, this is not the case (Figure 11). Although the single *PDZ* domain in *Carma1* and *Carma2* was vertically inherited, *Carma3* experienced both an expansion and contraction of its *PDZ* repertoire in the amniote ancestor. This expansion was due to two independent and highly-supported insertion events - one from *MPP4* and the other from *Carma1*. Parallel losses followed, with the copy from *MPP4* lost in birds and the copy from *Carma1* lost in mammals. As a result, even though all contemporary *Carma3*'s have the same domain architectures, the *PDZ* domain in chicken *Carma3* is paralogous, not orthologous, to the *PDZ* domain in mouse and human.

These case studies highlight the importance of using information from three levels of evolution - domain, gene, and species - when analyzing the history of a multidomain family. Without multiple species, the divergent history of the *PDZs* in *Carma3* in birds and mammals would not be apparent. Note also the pseudonode representing the ancestral *Carma2* in amniotes. This was the result of a loss of the *Carma2* gene in chicken. Our algorithm correctly identifies the missing *PDZ* in chicken as a gene loss, not a domain loss.

Conclusions

Here we propose a reconciliation-based framework that captures several aspects of multidomain evolution not represented in the Wagner and Dollo parsimony models that are widelyused in domain architecture analysis. Based on a model that considers domain sequence, as well as domain content, and incorporates an explicit model of events, our reconstruction algorithms are capable of inferring the correspondence between domain architecture evolution and gene and species evolution; the domain duplications, insertions, deletions and transfers that gave rise to present-day proteins; and the ancestral domain architectures from which they evolved.

As a demonstration of the power of a reconciliation-based approach, we presented an analysis of the domain shuffling in the multidomain Maguk family. This analysis uncovered multiple evolutionary scenarios that could not be reconstructed with Wagner parsimony, including parallel gains (the cassette of 3 *PDZs* in the *ZO*

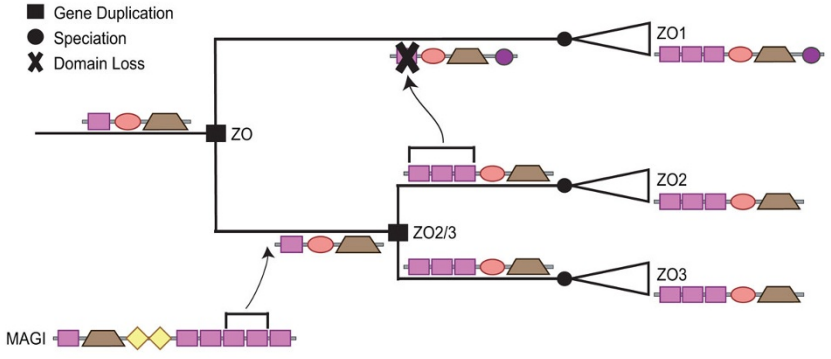

Figure 10 Reconstruction of PDZ domain shuffling in the ZO subfamily. Insertion of the two N-terminal *PDZ* domains in the common ancestor of *ZO2* and *ZO3*, followed by the insertion of the same two domains, plus the ancestral domain, from *ZO2* to *ZO1*. The ancestral architectures and domain insertions superimposed on the gene tree were reconstructed manually using events and associations of the ancestral nodes in the reconciled domain, gene, and species trees.

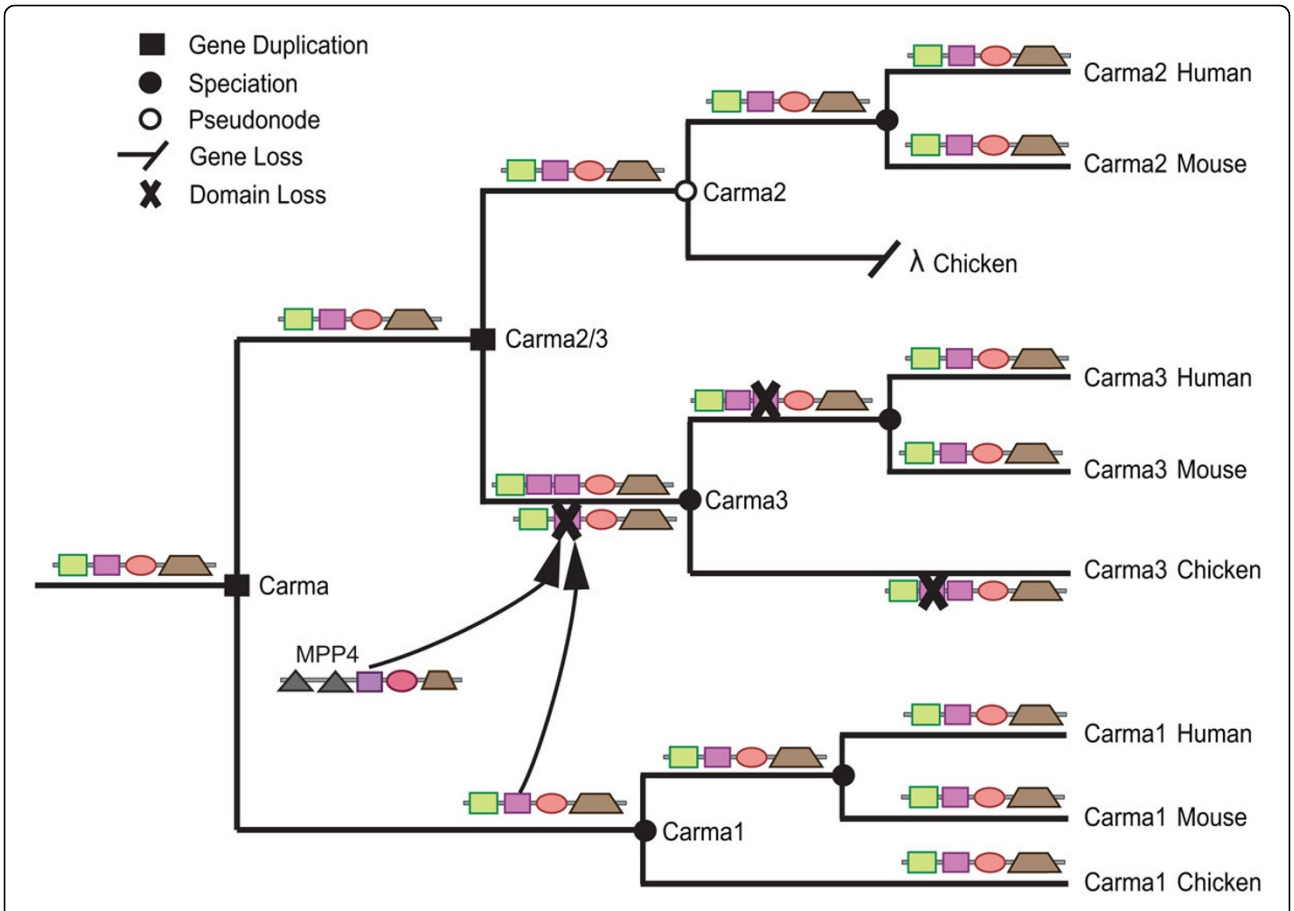

Figure 11 Reconstruction of PDZ domain shuffling in the Carma subfamily. Insertion of the *PDZ* domains from *MPP4* and *Carma1* into the ancestral *Carma3*. The two copies in *Carma3* subsequently experienced reciprocal loss - one domain was lost in the mammal ancestor and the other was lost in chicken. *Carma2* also experienced a gene loss in chicken. The ancestral architectures and domain insertions superimposed on the gene tree were reconstructed manually using events and associations of the ancestral nodes in the reconciled domain, gene, and species trees.

subfamily), parallel losses (*PDZ* in *Carma3*), and replacement of a domain with another domain from the same family (*L27* in the *MPP* subfamily). Further, our results suggest that some orthologous proteins with identical architectures nevertheless contain paralogous domains. This has intriguing implications for orthology-based function prediction.

Our analysis of the Maguks suggests that domain architectures may be much more plastic than previously thought. This contradicts prior studies suggesting that domain architectures rarely form more than once in evolutionary history, based on the argument that domain pairs that co-occur are selectively favorable and once united tend to persist [66]. However, if the same domain architectures were forming repeatedly, this pattern would not be discerned by Wagner parsimony. In contrast, our approach, which incorporates sequence information and an explicit event model, could reveal such patterns. To what extent is our understanding of multidomain evolution driven by the algorithms we use to study them? The results presented here argue for the reexamination of current theories of multidomain protein evolution using more information-rich approaches.

Our empirical study also reveals some of the challenges involved in applying this abstract approach to real data. Reconciliation-based event inference requires accurate domain and gene trees, yet mobile domains tend to be short and have low sequence conservation, making it particularly difficult to infer accurate trees for such domains. In our analysis, we developed an event support score that combines information from multiple phylogenetic hypotheses and focused on events with the strongest support. Development of rigorous approaches for accommodating uncertainty is an essential prerequisite for robust reconciliation-based analyses.

The reconciliation algorithms presented here are the first to consider the co-evolution of domains with both genes and species. A few other studies have considered

reconciliation using domain trees for the purpose of inferring domain content in ancestral species [27-29]. The optimization criteria used in those studies do not model domain shuffling events in individual families explicitly. It is therefore unclear whether they can be adapted to the problem of reconstructing the history of events in the evolution of multidomain protein families. Wu *et al*. [67] considered co-evolving domain trees using a model based on gene fusion and fission in a study of domain rearrangements in *Drosophila*. Their approach is inextricably linked to the problem of conserved sequence motif discovery and, similarly, is not well-suited to our question.

Expanding the model presented here is an important area for future work. Our algorithm reconstructs domain events within a single gene family. However, domain shuffling is likely also occurring across gene families. Further, many approaches to domain shuffling, including ours, make the implicit assumption that individual domain events are independent. Algorithms that capture the movement of multiple domains in a single event [68-70] are needed, as are algorithms that take domain order into account. Recent work integrating spatial information with phylogenetic reconciliation [71,72] is a promising direction for reconstructing domain order.

Probabilistic reconciliation models [73-78] are a particularly appealing direction for future work, because they require fewer simplifying assumptions, do not restrict the search space, can be instantiated with different evolutionary models to suit the needs of the data, and provide a context for formal testing of alternate hypotheses. Adapting this approach to the domain architecture context will be challenging. In contrast to gene and protein sequences, which are long strings of symbols from a small alphabet, domain architectures are short sequences from a very large alphabet and may not contain enough information to infer family-specific rates of insertion and deletion.

Methods

A reference gene tree and two domain trees were constructed for various components of the Maguk gene family. The gene tree was inferred using the non-functional guanylate kinase (*GuK*) domain, which is unique to this gene family. Domain trees were constructed for the two constituent domains that were analyzed in this study: the *L27* domains of the *MPP* subfamily and the *PDZ* domains, which are present in all Maguks except for the *CACNBs. Sequence acquisition:* An initial set of amino acid sequences of the *GuK, PDZ*, and *L27* domains from *Gallus gallus, Homo sapiens*, and *Mus musculus* was obtained from te Velthuis *et al*. [52]. This set was verified and expanded using the Uniprot Knowledge Base [79].

Sequence alignment: Multiple sequence alignments for the *GuK* and *PDZ* domains were constructed using MUSCLE [80], followed by extensive manual correction. The resulting alignments, with 293 and 107 amino acid positions, respectively, were trimmed in Tri- mAl [81] to remove sites with more than 70% gaps. The final alignments after trimming have 210 and 86 sites, respectively.

L27 domain alignments were constructed for the *L27-1* and *L27-2* subtypes individually using Expresso [82] under default parameters. Alignments were manually refined based on [61] and then merged using the "combine" function of T-Coffee [83]. These alignments were trimmed manually.

Phylogeny reconstruction: Phylogenetic trees for all domain and sequence families were reconstructed using maximum likelihood (ML) estimation. Model selection for all reconstructions was first performed on the trimmed alignment using Modelgenerator [84]. The best model for *PDZ* was LG+G, as supported by AIC1, AIC2, and BIC; the best for *GuK* was JTT+G, as supported by AIC2 and BIC (AIC1 supports JTT+I+G); and the best for *L27* was JTT.

ML trees for the *GuK*, *PDZ*, and *L27* domains were generated in PhyML [85] with 100 bootstraps, based on the selected best models. The *GuK* tree was rooted using the sequences of three functional guanylate kinase domains, from which the *GuK* domain is derived [86]. After rooting the *GuK* tree with these outgroups, one nearest neighbor interchange operation was performed on the least supported branch (bootstrap = 33), so that the *MPP5 GuKs* in all species form a monophyletic clade, instead of grouping some *MPP5* with *MAGI*. The *L27* tree was rooted based on the duplication that gave rise to *L27-1* and *L27-2*.

TreePuzzling: Support for the *L27* and *PDZ* domains was further evaluated by generating a 95% confidence set of trees based on the Expected Likelihood Weight method as implemented in CONSEL [87]. Tree-Puzzle [63] was used to sample 50,000 trees, whose likelihoods were assessed using the JTT model with 4 gamma rate categories. A consensus network was constructed from the *L27* confidence set using SplitsTree [64] with an edge threshold of 0.25.

Event support calculation: For the *PDZs*, the confidence set consists of 121 trees. To avoid spurious event inference, each tree was manually rooted, so as to separate *MPP* and *MAGI* from the other Maguk subfamilies, consistent with the *GuK* tree. Using the DE-DL model, each tree was then reconciled with the reconciled *GuK* tree, as described in the Results and Discussion section, using event costs $\kappa_D = 3.14$, $\kappa_I = 8.54$, and 2.72. All multiple optimal solutions were retained for each tree. This resulted in a total of 2358 reconciled trees. Event

support was calculated for each inferred domain insertion. The support value reflects the fraction of reconciled trees in which that particular event occurs.

For each insertion observed in at least one reconciliation, we calculate its support as follows: To ensure that each of the 121 trees has equal voting power, the multiple optimal solutions from each tree were averaged. Given a confidence set $\{T_D^i\}$ for domain D, the support value for insertion ε is

$$\frac{1}{|\{T_D^i\}|}\sum_i\sum_j \frac{I\left(\varepsilon,\ T_{DG}^{i,j}\right)}{|\left\{T_{DG}^{i,j}\right\}|}, \quad (1)$$

where $\left\{T_{DG}^{i,j}\right\}$ is the set of all optimal reconciliations of T_D^i and the indicator function $I\left(\varepsilon,\ T_{DG}^{i,j}\right)$ is equal to 1, if $T_{DG}^{i,j}$ contains an insertion that is identical to ε, and is equal to 0, otherwise. Domain insertions (d_1, d_2) and (d_1', d_2') in two different reconciled domain trees T_{DG} and T_{DG}' are considered to be identical if these four conditions are satisfied: (1) $L(T_{DG}(d_1)) = L\left(T_{DG}'\left(d_1'\right)\right)$, (2) $L(T_{DG}(d_2)) = L\left(T_{DG}'\left(d_2'\right)\right)$, (3) $M^{DG}(d_1) = M^{DG}\left(d_1'\right)$, and (4) $M^{DG}(d_2) = M^{DG}\left(d_2'\right)$, where $T_D(d)$ is the subtree rooted at d.

Additional material

Additional file 1: Additional file containing supplementary figures: stolzer32 supplemental.pdf Format: PDF

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The project was conceived and directed by MS and DD. MS, HL, and MX designed and implemented the algorithms. Empirical case studies carried out by KS, MX, and MS. MS, DD, MX, and HL wrote the manuscript. All authors read and approved the manuscript.

Acknowledgements
This work was supported by NSF grant DBI1262593, HFSP grant RGP0043/2013, and a Pittsburgh Supercomputing Center Computational Facilities Access Grant MCB000010P. The authors are deeply grateful to Annette McLeod for her help with the figures.

Declarations
Publication charges for this article were funded by NSF grant DBI1262593 to DD. This article has been published as part of *BMC Bioinformatics* Volume 16 Supplement 14, 2015: Proceedings of the 13th Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics: Bioinformatics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/16/S14.

Authors' details
[1]Department of Biological Sciences, Carnegie Mellon University, Forbes Ave, Pittsburgh, PA, 15213, USA. [2]School of Medicine, University of Pennsylvania, Curie Blvd, Philadelphia, PA, 19104, USA. [3]Department of Computer Science, Carnegie Mellon University, Forbes Ave, Pittsburgh, PA, 15213, USA.

Published: 2 October 2015

References
1. Moore A, Björklund A, Ekman D, Bornberg-Bauer E, Elofsson A: **Arrangements in the modular evolution of proteins.** *Trends Biochem Sci* 2008, **33(9)**:444-451.
2. Buljan M, Frankish A, Bateman A: **Quantifying the mechanisms of domain gain in animal proteins.** *Genome Biol* 2010, **11(7)**:74.
3. Basu M, Poliakov E, Rogozin I: **Domain mobility in proteins: functional and evolutionary implications.** *Brief Bioinform* 2009, **10(3)**:205-216.
4. Chothia C, Gough J: **Genomic and structural aspects of protein evolution.** *Biochem J* 2009, **419(1)**:15-28.
5. Finn R, Mistry J, Tate J, Coggill P, Heger A, *et al*: **The Pfam protein families database.** *Nucleic Acids Res* 2010, **38**:211-222.
6. Schultz J, Milpetz F, Bork P, Ponting C: **SMART, a simple modular architecture research tool: identification of signaling domains.** *PNAS* 1998, **95**:5857-5864.
7. Schultz J, Copley R, Doerks T, Ponting C, Bork P: **SMART: a web-based tool for the study of genetically mobile domains.** *Nucleic Acids Res* **28**:231-234.
8. Murzin A, Brenner S, Hubbard T, Chothia C: **SCOP: a structural classification of proteins database for the investigation of sequences and structures.** *J Mol Biol* 1995, **247(4)**:536-40.
9. Apweiler R, Attwood T, Bairoch A, Bateman A, Birney E, *et al*: **The InterPro database, an integrated documentation resource for protein families, domains and functional sites.** *Nucleic Acids Res* 2001, **29(1)**:37-40.
10. Marchler-Bauer A, Anderson J, Cherukuri P, DeWeese-Scott C, Geer L, *et al*: **CDD: a Conserved Domain Database for protein classification.** *Nucleic Acids Res* 2005, **33(Database)**:192-6.
11. Karev G, Wolf Y, Berezovskaya F, Koonin E: **Gene family evolution: an in-depth theoretical and simulation analysis of non-linear birth-death-innovation models.** *BMC Evol Biol* 2004, **4**:32.
12. Tordai H, Nagy A, Farkas K, Banyai L, Patthy L: **Modules, multidomain proteins and organismic complexity.** *FEBS J* 2005, **272(19)**:5064-5078.
13. Ye Y, Godzik A: **Comparative analysis of protein domain organization.** *Genome Res* 2004, **14(3)**:343-353.
14. Bashton M, Chothia C: **The geometry of domain combination in proteins.** *J Mol Biol* 2002, **315(4)**:927-939.
15. Weiner J, Beaussart F, Bornberg-Bauer E: **Domain deletions and substitutions in the modular protein evolution.** *FEBS J* 2006, **273(9)**:2037-2047.
16. Vogel C, Teichmann S, Pereira-Leal J: **The relationship between domain duplication and recombination.** *J Mol Biol* 2005, **346(1)**:355-365.
17. Karev G, Wolf Y, Rzhetsky A, Berezovskaya F, Koonin E: **Birth and death of protein domains: a simple model of evolution explains power law behavior.** *BMC Evol Biol* 2002, **2(1)**:18.
18. Marcotte E, Pellegrini M, Ng H, Rice D, Yeates T, Eisenberg D: **Detecting protein function and protein-protein interactions from genome sequences.** *Science* 1999, **285(5428)**:751-753.
19. Basu M, Carmel L, Rogozin I, Koonin E: **Evolution of protein domain promiscuity in eukaryotes.** *Genome Res* 2008, **18(3)**:449-461.
20. Cohen-Gihon I, Fong J, Sharan R, Nussinov R, Przytycka T, Panchenko A: **Evolution of domain promiscuity in eukaryotic genomes-a perspective from the inferred ancestral domain architectures.** *Mol Biosyst* 2011, **7(3)**:784-792.
21. Forslund K, Henricson A, Hollich V, Sonnhammer E: **Domain tree-based analysis of protein architecture evolution.** *Mol Biol Evol* 2008, **25(2)**:254-264.
22. Fong J, Geer L, Panchenko A, Bryant S: **Modeling the evolution of protein domain architectures using maximum parsimony.** *J Mol Biol* 2007, **366(1)**:307-315.
23. Kummerfeld S, Teichmann S: **Protein domain organisation: adding order.** *BMC Bioinformatics* 2009, **10**:39.
24. Snel B, Bork P, Huynen M: **Genomes in flux: the evolution of archaeal and proteobacterial gene content.** *Genome Res* 2002, **12**:17-25.
25. Björklund A, Ekman D, Light S, Frey-Skött J, Elofsson A: **Domain rearrangements in protein evolution.** *J Mol Biol* 2005, **353(4)**:911-923.
26. Przytycka T, Davis G, Song N, Durand D: **Graph theoretical insights into evolution of multidomain proteins.** *J Comput Biol* 2006, **13(2)**:351-363.

27. Behzadi B, Vingron M: **Reconstructing domain compositions of ancestral multi-domain proteins.** In *Comparative Genomics LNCS* Bourque, G., El-Mabrouk, M 2006, **4205**:1-10.
28. Wiedenhoeft J, Krause R, Eulenstein O: **Inferring evolutionary scenarios for protein domain compositions.** *Bioinformatics Research and Applications LNCS* 2010, **6053**:179-190.
29. Homilius M, Wiedenhoeft J, Thieme S, Kel I, *et al*: **Cocos: Constructing multi-domain protein phylogenies.** *PLoS Curr* 2011, **3**:1240.
30. Goodman M, Czelusniak J, Moore G, Romero-Herrera A, Matsuda G: **Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences.** *Syst Zool* 1979, **28**:132-163.
31. Page R, Charleston M: **Reconciled trees and incongruent gene and species trees.** *Mathematical Heirarchies and Biology, DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 1996, **37**:57-70.
32. Hallett M, Lagergren J, Tofigh A: **Simultaneous identification of duplications and lateral transfers.** *RECOMB 2004: Proceedings of the Eigth International Conference on Research in Computational Biology* ACM Press, New York, NY, USA; 2004, 347-356.
33. Nakhleh L, Ruths D, Innan H: **Gene trees, species trees, and species networks.** In *Meta-analysis and Combining Information in Genetics and Genomics.* CRC Press, Boca Raton, FL, USA;Guerra, R., Goldstein, D 2009:275-293.
34. Nakhleh L: **Evolutionary phylogenetic networks: models and issues.** In *The Problem Solving Handbook for Computational* Heath, L., Ramakrishnan, N 2010, 125-158.
35. Doyon J, Ranwez V, Daubin V, Berry V: **Models, algorithms and programs for phylogeny reconciliation.** *Brief Bioinform* 2011, **12**:392-400.
36. Tofigh A, Hallett M, Lagergren J: **Simultaneous identification of duplications and lateral gene transfers.** *TCBB* 2011, **8**:517-535.
37. David L, Alm E: **Rapid evolutionary innovation during an Archaean genetic expansion.** *Nature* 2011, **469**:93-96.
38. Stolzer M, Lai H, Xu M, Sathaye D, Durand D: **Inferring duplications, losses, transfers, and incomplete lineage sorting with non-binary species trees.** *Bioinformatics* 2012, **28**:409-415.
39. Bansal M, Alm E, Kellis M: **Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss.** *Bioinformatics* 2012, **28**:283-291.
40. Huson D, Scornavacca C: **A survey of combinatorial methods for phylogenetic networks.** *Genome Biol Evol* 2011, **3**:23-35.
41. Donati B, Baudet C, Sinaimeri B, Crescenzi P, Sagot M.-F: **EUCALYPT: efficient tree reconciliation enumerator.** *Algorithms for Molecular Biology* 2015, **10(1)**:11, doi:10.1186/s13015-014-0031-3.
42. Song N, Joseph J, Davis G, Durand D: **Sequence similarity network reveals common ancestry of multidomain proteins.** *PLoS Comput Biol* 2008, **4**:1000063.
43. Marcotte E, Pellegrini M, Thompson M, Yeates T, Eisenberg D: **A combined algorithm for genome-wide prediction of protein function.** *Nature* 1999, **402(6757)**:83-86.
44. Patthy L: **Intron-dependent evolution: preferred types of exons and introns.** *FEBS Lett* 1987, **214(1)**:1-7.
45. Sayah D, Sokolskaja E, Berthoux L, Luban J: **Cyclophilin A retrotransposition into TRIM5 explains owl monkey resistance to HIV-1.** *Nature* 2004, **430(6999)**:569-573.
46. Long M, Betran E, Thornton K, Wang W: **The origin of new genes: glimpses from the young and old.** *Nat Rev Genet* 2003, **4(11)**:865-75.
47. Vinckenbosch N, Dupanloup I, Kaessmann H: **Evolutionary fate of retroposed gene copies in the human genome.** *PNAS* 2006, **103(9)**:3220-3225.
48. Jones C, Custer A, Begun D: **Origin and evolution of a chimeric fusion gene in *Drosophila subobscura, D. madeirensis* and *D. guanche*.** *Genetics* 2005, **170(1)**:207-219.
49. Stolzer M: **Phylogenetic inference for multidomain proteins.** PhD thesis, Carnegie Mellon University, Pittsburgh, PA; 2012, Aug.
50. Vernot B, Stolzer M, Goldman A, Durand D: **Reconciliation with non-binary species trees.** *J Comput Biol* 2008, **15**:981-1006.
51. Cormen T, Leiserson C, Rivest R: **Introduction to Algorithms.** 1990.
52. te Velthuis A, Admiraal J, Bagowski C: **Molecular evolution of the MAGUK family in metazoan genomes.** *BMC Evol Biol* 2007, **7**:129.
53. Mendoza A, Suga H, Ruiz-Trillo I: **Evolution of the MaGuK protein gene family in premetazoan lineages.** *BMC Evol Biol* 2010, **10**:93.
54. Caruana G: **Genetic studies define MAGUK proteins as regulators of epithelial cell polarity.** *Int J Dev Biol* 2002, **46**:511-518.
55. Stucke V, Timmerman E, Vandekerckhove J, Gevaert K, Hall A: **The MAGUK protein MPP7 binds to the polarity protein hDlg1 and facilitates epithelial tight junction formation.** *Mol Biol Cell* 2007, **18**:1744-1755.
56. Funke L, Dakoji S, Bredt D: **Membrane-associated guanylate kinases regulate adhesion and plasticity at cell junctions.** *Annu Rev Biochem* 2005, **74**:219-245.
57. Elias G, Nicoll R: **Synaptic trafficking of glutamate receptors by maguk scaffolding proteins.** *Trends Cell Biol* 2007, **17**:343-352.
58. Emes R, Pocklington A, Anderson C, Bayes A, Collins M, Vickers C, Croning M, Malik B, Choudhary J, Armstrong J, Grant S: **Evolutionary expansion and anatomical specialization of synapse proteome complexity.** *Nat Neurosci* 2008, **11**:799-806.
59. Good M, Zalatan J, Lim W: **Scaffold proteins: hubs for controlling the flow of cellular information.** *Science* 2011, **332**:680-686.
60. McGee AW, Dakoji SR, Olsen O, Bredt DS, Lim WA, Prehoda KE: **Structure of the SH3-guanylate kinase module from PSD-95 suggests a mechanism for regulated assembly of MAGUK scaffolding proteins.** *Mol Cell* 2001, **8**:1291-1301.
61. Feng W, Long J, Fan J, Suetake T, Zhang M: **The tetrameric L27 domain complex as an organization platform for supramolecular assemblies.** *Nat Struct Mol Biol* 2004, **11**:475-480.
62. Shimodaira H, Hasegawa M: **CONSEL: for assessing the confidence of phylogenetic tree selection.** *Bioinformatics* 2001, **17**:1246-1247.
63. Schmidt HA, Strimmer K, Vingron M, von Haeseler A: **TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing.** *Bioinformatics* 2002, **18**:502-504.
64. Huson D, Bryant D: **Application of phylogenetic networks in evolutionary studies.** *Mol Biol Evol* 2006, **23(2)**:254-267.
65. Yang X, Xie X, Chen L, Zhou H, Wang Z, Zhao W, *et al*: **Structural basis for tandem L27 domain-mediated polymerization.** *FASEB J* 2010, **24**:4806-4815.
66. Gough J: **Convergent evolution of domain architectures (is rare).** *Bioinformatics* 2005, **21(8)**:1464-1471.
67. Wu Y, Rasmussen M, Kellis M: **Evolution at the subgene level: domain rearrangements in the *Drosophila* phylogeny.** *Mol Biol Evol* 2012, **29**:689-705.
68. Björklund A, Light S, Sagit R, Elofsson A: **Nebulin: a study of protein repeat evolution.** *J Mol Biol* 2010, **402(1)**:38-51.
69. Vogel C, Berzuini C, Bashton M, Gough J, Teichmann S: **Supra-domains: evolutionary units larger than single protein domains.** *J Mol Biol* 2004, **336(3)**:809-823.
70. Han J, Batey S, Nickson A, Teichmann S, Clarke J: **The folding and evolution of multidomain proteins.** *Nat Rev Mol Cell Biol* 2007, **8**:319-330.
71. Tremblay Savard O, Bertrand D, El-Mabrouk N: **Evolution of orthologous tandemly arrayed gene clusters.** *BMC Bioinformatics* 2011, **12(Suppl 9)**:2.
72. Be´rard S, Gallien C, Boussau B, Szöllsi GJ, Daubin V, Tannier E: **Evolution of gene neighborhoods within reconciled phylogenies.** *Bioinformatics* 2012, **28(18)**:382-388.
73. Liu L, Pearl D: **Species trees from gene trees: reconstructing bayesian posterior distributions of a species phylogeny using estimated gene tree distributions.** *Syst Biol* 2007, **56(3)**:504-514.
74. Akerborg O, Sennblad B, Arvestad L, Lagergren J: **Simultaneous Bayesian gene tree reconstruction and reconciliation analysis.** *PNAS* 2009, **106**:5714-5719.
75. Arvestad L, Berglund A, Lagergren J, Sennblad B: **Bayesian gene/species tree reconciliation and orthology analysis using MCMC.** *Bioinformatics* 2003, **19(Suppl 1)**:7-15.
76. Arvestad L, Berglund A, Lagergren J, Sennblad B: **Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution.** *RECOMB 2004: Proceedings of the Eigth International Conference on Research in Computational Biology* ACM Press. ACM, San Diego, California, USA; 2004, 326-335.
77. Rasmussen M, Kellis M: **Unified modeling of gene duplication, loss, and coalescence using a locus tree.** *Genome Res* 2012.
78. Go´recki P, Burleigh G, Eulenstein O: **Maximum likelihood models and algorithms for gene tree evolution with duplications and losses.** *BMC Bioinformatics* 2011, **12(Suppl 1)**:15.
79. **Uniprot Consortium: Uniprot: a hub for protein information.** *Nucleic Acids Res* 2015, **43**:204-212.

80. Edgar R: **MUSCLE: multiple sequence alignment with high accuracy and high throughput.** *Nucleic Acids Res* 2004, **32**:1792-1797.
81. Capella-Gutie´rrez S, Silla-Martínez J, Gabaldo´ n T: **trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses.** *Bioinformatics* 2009, **25**:1972-1973.
82. Armougom F, Moretti S, Poirot O, Audic S, Dumas P, Schaeli B, Keduas V, Notredame C: **Expresso: automatic incorporation of structural information in multiple sequence alignments using 3d-coffee.** *Nucleic acids research* 2006, **34(suppl 2)**:604-608.
83. Notredame C, Higgins D, Heringa J: **T-Coffee: A novel method for fast and accurate multiple sequence alignment.** *J Mol Biol* 2000, **302(1)**:205-17.
84. Keane T, Creevey C, Pentony M, Naughton T, McInerney J: **Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified.** *BMC Evol Biol* 2006, **6(29)**.
85. Guindon S, Dufayard J, Lefort V, Anisimova M, Hordijk W, Gascuel O: **New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0.** *Syst Biol* 2010, **59**:307-321.
86. Olsen O, Bredt DS: **Functional analysis of the nucleotide binding domain of membrane-associated guanylate kinases.** *Journal of Biological Chemistry* 2003, **278(9)**:6873-6878.
87. Schmidt HA: **Testing tree topologies.** In *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing.* Cambridge University Press, Cambridge, UK;Lemey, P., Salemi, M., Vandamme, A.-M 2009:381-396, Chap. 12.

doi:10.1186/1471-2105-16-S14-S8
Cite this article as: Stolzer *et al.*: Event inference in multidomain families with phylogenetic reconciliation. *BMC Bioinformatics* 2015 **16**(Suppl 14):S8.