Evolution at the Subgene Level: Domain Rearrangements in the *Drosophila* Phylogeny

Yi-Chieh Wu,*[,1] Matthew D. Rasmussen,[1] and Manolis Kellis*[,1,2]

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
[2]Broad Institute of MIT and Harvard

***Corresponding author:** E-mail: yjw@mit.edu; manoli@mit.edu.

Associate editor: John Parsch

Abstract

Although the possibility of gene evolution by domain rearrangements has long been appreciated, current methods for reconstructing and systematically analyzing gene family evolution are limited to events such as duplication, loss, and sometimes, horizontal transfer. However, within the *Drosophila* clade, we find domain rearrangements occur in 35.9% of gene families, and thus, any comprehensive study of gene evolution in these species will need to account for such events. Here, we present a new computational model and algorithm for reconstructing gene evolution at the domain level. We develop a method for detecting homologous domains between genes and present a phylogenetic algorithm for reconstructing maximum parsimony evolutionary histories that include domain generation, duplication, loss, merge (fusion), and split (fission) events. Using this method, we find that genes involved in fusion and fission are enriched in signaling and development, suggesting that domain rearrangements and reuse may be crucial in these processes. We also find that fusion is more abundant than fission, and that fusion and fission events occur predominantly alongside duplication, with 92.5% and 34.3% of fusion and fission events retaining ancestral architectures in the duplicated copies. We provide a catalog of ~9,000 genes that undergo domain rearrangement across nine sequenced species, along with possible mechanisms for their formation. These results dramatically expand on evolution at the subgene level and offer several insights into how new genes and functions arise between species.

Key words: phylogenetics, gene fusion and fission, domain and architecture evolution.

Research article

Introduction

Evolution can change the structure and function of genes in many ways. For example, gene duplication has long been identified as a major mechanism for generating new genes and functions (Ohno 1970; Lynch and Conery 2000; Long et al. 2003), whereas gene loss plays a similarly important role in shaping genomic content (Hahn, Demuth, et al. 2007; Niimura and Nei 2007). These events, as well as several others such as horizontal gene transfer, gene conversion, and domain rearrangement, interact together to generate "gene families," clusters of orthologous and paralogous genes with detectable common ancestry. By studying the genetic sequences of a family, one can infer many of the evolutionary events likely responsible for its creation.

The history of a gene family is often represented by two trees: the "gene tree," which describes the evolutionary relationship of the genes, and the "species tree," which describes the relationship of the species. The gene tree can be thought of as evolving "inside" of the species tree (fig. 1*B*). In the simplest case, these two trees are congruent (share the same topology), indicating that all the genes of the family are orthologs. However, if the two trees differ, then events such as gene duplication and loss have occurred. One can infer these events by combining several computational methods. Phylogenetic methods, such as maximum likelihood (Felsenstein 1981) or neighbor joining (Saitou and Nei 1987), can be used to reconstruct a gene tree and species tree from molecular sequences, and special algorithms called "recon- ciliation methods" (Goodman et al. 1979; Page 1994; Chen et al. 2000) can be used to determine how the gene tree fits inside, or rather "reconciles," to the species tree. Lastly, it is the reconciliation that indicates the particular number and order of evolutionary events that have occurred in the gene family.

With the growing availability of genome sequences, this phylogenetic analysis can be applied across both sizable clades and whole genomes in a research field called "phylogenomics" (Eisen 1998; Eisen and Fraser 2003). Many computational methods have been developed for detecting and reconstructing gene families as well as their events (Lynch and Conery 2000; Zmasek and Eddy 2002; Hahn et al. 2005; Rasmussen and Kellis 2007, 2011; Wapinski et al. 2007, Arvestad et al. 2009; Butler et al. 2009; Vilella et al. 2009). This has led to a better understanding of how evolution shapes the gene content of many different species such as prokaryotes (David and Alm 2011), yeasts (Wapinski et al. 2007, Butler et al. 2009), flies (Hahn, Demuth, et al. 2007), and vertebrates (Vilella et al. 2009).

Despite the sophisticated underlying models in these methods, a common assumption is to consider a gene as evolving as a single unit. However, duplications, losses, and other events can occur at the subgene level, and it has been suggested that homology inference be applied to domains rather than proteins (Ponting and Russell 2002).

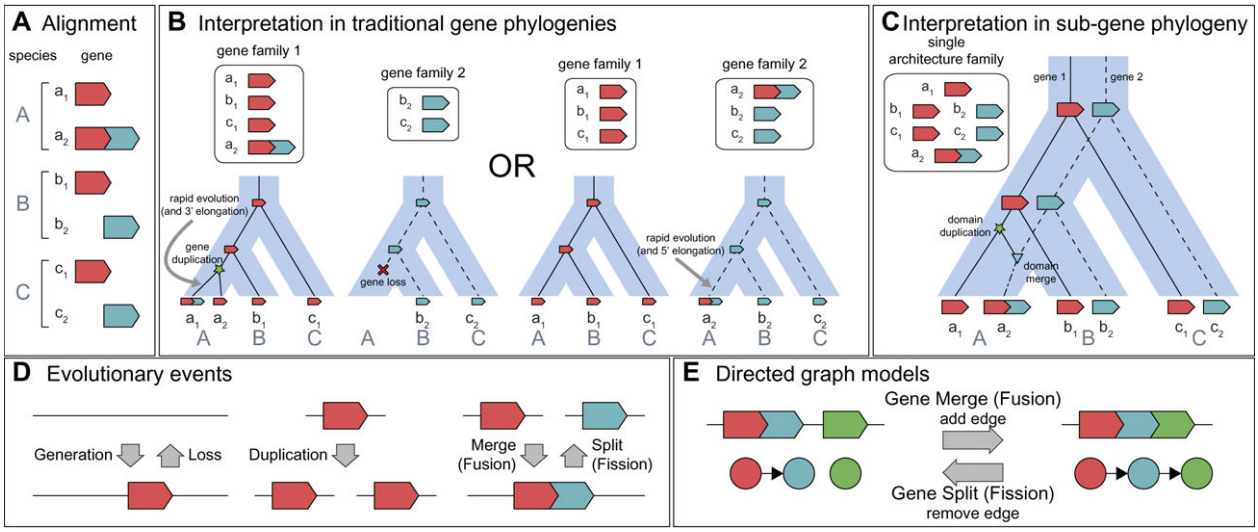

FIG. 1. Relationship between species trees, gene trees, and architecture scenarios. (*A*) Gene sequences are compared across species, and a multiple sequence alignment is constructed. Due to the presence of domains or complicated evolutionary mechanisms, these alignments may have a block structure indicating similarity at the subgene level. (*B*) In conventional phylogenetics, genes that descend from a single common ancestor are clustered into a gene family, and the history of gene families are viewed through gene trees (black lines) that evolve inside a species tree (blue area). Duplication ($\star$), loss ($\times$), and speciation (colored subgene blocks) events are inferred through the reconcilation of gene trees to species trees. Since each gene can belong to only a single gene family, joint histories that are evident from the architecture structure cannot be captured. (*C*) In subgene phylogenetics as presented in this work, a gene family is generalized to an architecture family in order to capture the relationships between genes with shared modules. This allows the reconstruction of gene histories to be architecture aware, with an architecture scenario depicting more complicated events such as merges ($\bigtriangledown$) and splits (not shown). By definition, architecture scenarios use a known species tree, with architectures evolving from a parent species to a child species; thus, no reconcilation is required, and speciation events are not modeled. In this example, the joint histories of the red and teal modules are determined, including their recent merge in the branch leading to species A, corresponding to the formation of chimeric gene a_2. (*D*) We allow for five types of evolutionary events, two (merge and split) of which are not typically captured in conventional gene phylogenetics. (*E*) Gene architectures are modeled using directed graphs, with nodes representing modules and edges representing neighboring modules (within the same gene). Rearrangements of these graphs correspond to evolutionary events: Adding or removing nodes correspond to generation, duplication, or loss events (not shown), and adding or removing edges correspond to merge or split events.

Additionally, events such gene fusion and fission challenge the current definition of a gene family, as they can form genes that have varying phylogeny and homology across the gene sequence. These more complicated events could play very important roles in generating novel genes and functions, as they are the primary source of new domain architectures that are thought to be a main source of biological complexity in the human genome and other species (Yanai et al. 2002; Pasek et al. 2006).

There are already several experimentally discovered examples of fusion and fission events. For example, *jingwei* is a chimeric gene found in *Drosophila yakuba* that arose through the fusion of the two genes *yande* (involved in nuclear mRNA splicing) and *alcohol dehydrogenase* (*Adh*). Although a fusion of genes is likely deleterious, several factors in this case have contributed to *jingwei*'s retention. First, the ancestral functions involved in this fusion event were kept intact, as *yande* is itself a recent duplicate of *yellow-emperor*, and the *Adh* portion of *jingwei* is a retrotransposed copy of *Adh* (Long and Langley 1993; Long et al. 1999; Wang et al. 2000). This allowed the *jingwei* to acquire a novel function in more specific binding for long-chain alcohols (Shih and Jones 2008). Second, *jingwei* has inherited the promoter sequence of *yande*, preventing degeneration of the retrotransposed *Adh* into a pseudogene. Other examples of gene fusion events in *Drosophila* gave rise to *Adh–Twain* (Jones et al. 2005), *Adh–Finnegan* (Jones and Begun 2005), *siren* (Shih and Jones 2008), *sphinx* (Wang et al. 2002), and *Quetzalcoatl* (Rogers et al. 2010), which have diverse functions in metabolic processes and male courtship behavior. In addition, studies have identified fusion and fission events within clades such as bacteria (Suhre and Claverie 2004; Pasek et al. 2006) and fungi (Durrens et al. 2008), and specific chimeric genes have been studied in humans (Thomson et al. 2000; Courseaux and Nahon 2001) and plants (Wang et al. 2006). However, although intron phase correlations suggest that as many as ~19% of exons in eukaryotic genes might have been formed by exon shuffling (Long et al. 1995), large-scale methods for the systematic identification and reconstruction of domain evolution and gene fusion and fission events are still lacking.

Though they do not reconstruct the history of these events, many directed studies have analyzed domain rearrangements in search of functional or evolutionary insights (Bornberg-Bauer et al. 2005; Moore et al. 2008). Quantitative analyses have shown that fusions are more prevalent to fission (Snel et al. 2000), that the number of neighbors per domain follows a power law (Apic et al. 2001, 2003) (though

this could be attributed to limited coverage; Han et al. 2005), and that specific domain combinations are more conserved than would be expected from random domain shuffling (Apic et al. 2003). Also, sequence similarity networks have been used to determine gene families of multidomain proteins (Enright et al. 2002; Uchiyama 2006; Song et al. 2008), mechanisms of domain deletions, shufflings, and substitutions have been proposed (Weiner and Bornberg-Bauer 2006; Weiner et al. 2006), and protein interaction maps have been generated based on gene fusions (Enright et al. 1999; Enright and Ouzounis 2001).

More recently, phylogenomic methods have been developed to handle gene fusion and fission events or domain evolution, with initial approaches discovering domains de novo through sequence similarity (Snel et al. 2000) and later methods shifting to rely on underlying domain models using databases such as InterPro (Hunter et al. 2009), Pfam (Bateman et al. 2002), SCOP (Murzin et al. 1995), SMART (Schultz et al. 1998), and CDD (Marchler-Bauer et al. 2005). These studies focused on widely divergent species spanning all three domains of life and make three types of simplifying assumptions: 1) Only the presence or absence of architectures in complete genomes are considered, with both architecture count and sequence information ignored (Gough 2005; Kummerfeld and Teichmann 2005; Fong et al. 2007), 2) copy numbers for architectures are considered but domain ordering is ignored, and the models have leaned towards theoretic formulations and only been applied to a limited amount of biological data (Behzadi and Vingron 2006; Przytycka et al. 2006; Wiedenhoeft et al. 2011), and 3) domain level events are mapped onto existing gene trees, with agreement between evolutionary events considered only after the independent mappings (Forslund et al. 2008).

Our work continues along these recent methods in extending phylogenomics from genes to subgene domains. We present the first phylogenomic approach that combines de novo discovery of subgene evolutionary units (which we term as "modules"), a general model of gene evolution that captures module gain, loss, duplication, and rearrangement, and a phylogenetic reconstruction algorithm that simultaneously traces the history of all modules while taking into account a common species tree topology. By focusing on modules, we are in many ways looking at how new genes are generated. That is, we can consider gene generation at a very low level through mutations and insertion/deletions or at a very high level through gene duplication and loss. This work proposes a middle perspective that looks at gene generation through the generation of new modules and the duplication, loss, and rearrangement of existing modules.

This paper presents three distinct contributions to subgene phylogenomics:

- We present a method for identifying homologous modules for a family of closely related species. Our approach uses sequence similarity to define modules as the basic unit of inheritance and therefore is not limited to existing domain databases, which may be biased towards domains with known structures or domains found in well-studied proteins. We show that the resulting modules are biologically meaningful; in particular, they are frequently produced through exon shuffling events, and, when such annotations are available, they tend to keep functional domains as a single unit.
- We develop a model for gene evolution that captures architecture rearrangements, which we define as module generation, duplication, loss, merge (fusion), and split (fission) events (fig. 1*D*). In contrast to many previous phylogenetic approaches, our model traces "gene evolution" rather than "architecture evolution," allowing us to explicitly capture module duplications and parallel merges and splits.
- We present a maximum parsimony algorithm, Species Tree informed Architecture Reconstruction—Maximum Parsimony (STAR-MP), for inferring module architecture evolution based on (reconstructed) module phylogenies, extant module architectures, and a known species tree. Along with our evolutionary model, this algorithm is less restrictive than previous phylogenetic approaches, retaining the advantages of each. In particular, we assume a known species tree, as the added information can improve gene tree reconstruction; we do not rely on a reference gene or domain but instead view modules as the primary unit of genes, allowing us to trace the evolutionary history of genes related through any subsequence within a single reconstruction; we incorporate sequence information for each module captured through phylogenetic reconstruction; and we consider the statistical support of our reconstructions through bootstrapping. The STAR-MP software is available for download at http://compbio.mit.edu/starmp/.

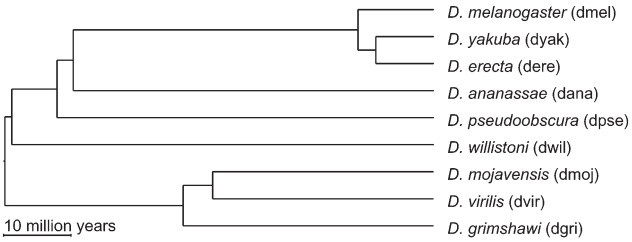

FIG. 2. Species and phylogeny of the *Drosophila* clade. The phylogeny of nine *Drosophila* species used in our analysis, as estimated by Tamura et al. (2004).

To demonstrate the sensitivity and robustness of our methods, we consider eukaryotic species that are evolutionary closely related, where a species tree is well supported and horizontal gene transfer is unlikely and not modeled. We also consider the problem of detecting architecture rearrangements at a smaller timescale, identifying only merge and split events that have occurred in recent history; we focus our analysis on the *Drosophila* clade (fig. 2), as it has a dense phylogeny, a relatively recent (~60 My old) history (Hahn, Han, et al. 2007), and includes both close and distant species. Furthermore, at least 47 putative chimeric genes have been identified within *D. melanogaster* (Zhou et al. 2008; Rogers et al. 2009), and it has been estimated that ~30% of the new genes in the *D. melanogaster* species

subgroup are chimeric (Zhou et al. 2008). We have used our methods to trace the complete history of all genes through their modules in nine *Drosophila* species and report numerous striking examples of architecture evolution that cannot be captured by traditional gene-level methods.

Materials and Methods

Genomic Sequences and Species Phylogeny

Analysis was performed on nine species within the *Drosophila* genus: *D. melanogaster*, *D. yakuba*, *D. erecta*, *D. ananassae*, *D. pseudoobscura*, *D. willistoni*, *D. mojavensis*, *D. virilis*, and *D. grimshawi*. Sequences were obtained from FlyBase (May 2009 release), and we analyzed the longest protein sequence per gene and assumed a known species tree (Tamura et al. 2004) (fig. 2).

Definitions

Due to fusion and fission, a gene may contain specific domains (or more generally DNA segments) whose evolutionary history differs from the rest of the gene. Therefore, we introduce several new concepts to describe the possible relationships between such genes. Our primary unit of evolution is the module, which is a gene subsequence inherited as a single unit without internal rearrangements or breaks across the species under comparison. Modules discovered from sequence similarity are distinct from structural or functional domains of a protein, though, as we will show, they often agree. Each gene may contain one or more nonoverlapping modules. These modules may share homology with other modules present within the same gene or in other genes. We call a cluster of homologous modules a "module family," defined as the set of modules that descend from a single ancestral module in the last common ancestor (LCA) of all species under consideration.

For each gene, we define its "architecture" as the ordered list of modules it contains. Each species contains a set of genes, which corresponds to a multiset of architectures. We generalize the concept of a gene family to that of a "(gene) architecture family," which contains the maximal set of genes connected by module homology. Whereas the evolutionary histories of gene families are represented by gene trees, the histories of architecture families are represented by "architecture directed acyclic graphs" (DAGs), which extend gene trees by capturing module generation, fusion, and fission events, in addition to module duplication and loss. Lastly, we define an "architecture scenario" as the multiset of ancestral architectures and evolutionary events mapped onto a known species tree, where each species tree node shows the type and copy number of architectures it contains, and each species tree branch shows the events that have occurred along that branch. In reconstructing architecture scenarios, we will assume a known species tree and infer ancestral architectures and events without requiring a reconciliation mapping. All trees within this work are rooted phylogenetic trees, in which the leaf nodes represent extant evolutionary objects (e.g., extant species or modules in extant species) and the internal nodes represent ancestral objects (e.g., ancestral species or ancestral modules in ancestral species).

Our model for architecture evolution allows for the following evolutionary events: "generation," in which a new module is created; "duplication," in which an existing module is duplicated; "loss," in which an existing module is lost; "merge," in which two modules that appeared at the ends of two separate architectures are joined as neighbors in a single gene; and "split," in which two modules that appeared as neighbors in a single gene are split and appear at the ends of two separate genes. We also make the further assumption that a module can be generated at most once. This is similar to the assumption used in Dollo parsimony, in which a single generation in the LCA followed by (multiple) losses is more likely than multiple independent generation events. We represent an architecture as a DAG capturing module ordering relationships between consecutive modules. Each evolutionary event corresponds to a simple graph operation (fig. 1*E*), and determining architecture rearrangements becomes a matter of graph rearrangements using these operations (supplementary section 1, Supplementary Material online).

Architecture-Aware Phylogenomic Pipeline

We present a novel phylogenomic pipeline for the architecture-aware reconstruction of gene evolution (fig. 3). The pipeline has three main stages: 1) identifying modules and module families from the genomic sequences, 2) clustering architectures into architecture families, and 3) reconstructing architecture scenarios from the architecture families and the known species tree.

Identifying Modules and Module Families

To identify modules and their boundaries, we ran pairwise all versus all BLASTp comparisons (Altschul et al. 1997) between the species' proteomes, discarding any BLAST hit with e value $>1 \times 10^{-5}$ or percent identity $<60\%$. The remaining alignments were extended using LALIGN (Huang and Miller 1991), and the best hit between each query and subject pair was retained. These were refiltered by e value and percent identity, and short alignments (<50 aa) and promiscuous hits (genes with >80 hits) were removed. A list of potential module boundaries was then found using the residue correlation matrix as in the ADDA algorithm (Heger and Holm 2003) (resolution = 10 aa, minimum module length = 30 aa), and boundaries within 30 aa of a LALIGN alignment boundary were retained. The resulting module instances were clustered into module families through OrthoMCL with default parameters (Enright et al. 2002), where the nodes represent module instances and edges are weighted by the bit score of the LALIGN hit multiplied by the relative overlap of the modules.

Note that if desired, these steps can be replaced by matching gene sequences against a database of known structural or functional protein domains to simultaneously detect the domain boundaries and domain families. However, our approach is more general as it defines modules as evolutionarily conserved units without relying on previ-

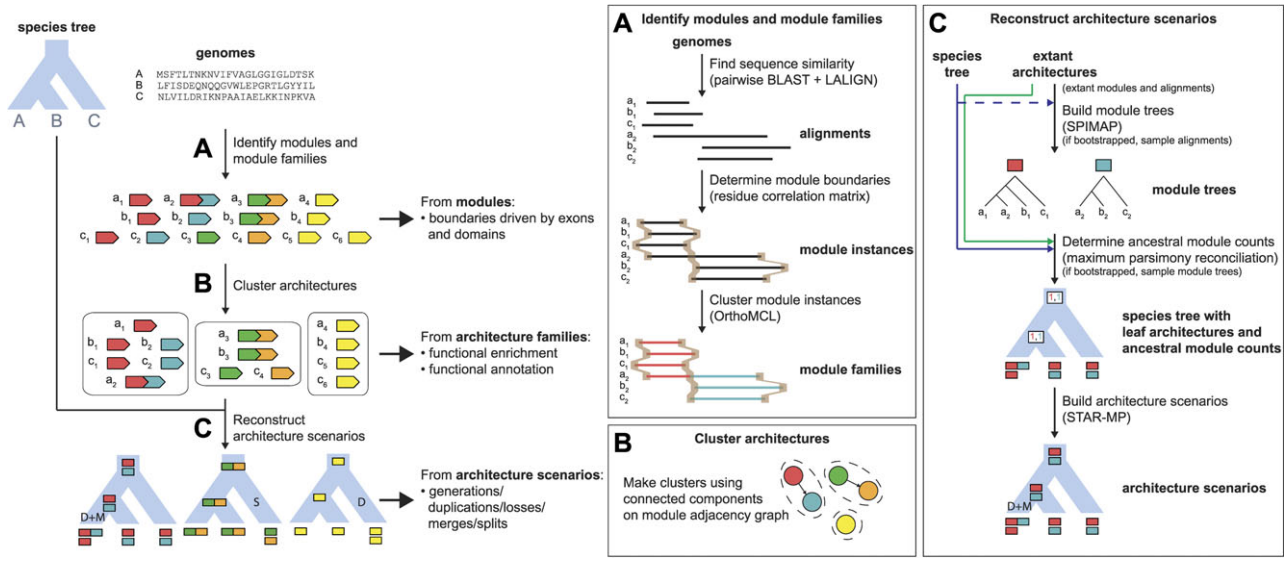

FIG. 3. Overview of our phylogenomic pipeline. At left, the pipeline is separated into three main stages and takes as input the set of all gene sequences across several species and the known species tree relating the species. (*A*) In the first stage, gene sequences are compared across species, module boundaries are found, and modules are clustered according to similarity, resulting in a set of homologous module families. (*B*) In the second stage, a module adjacency graph is constructed based on these module families, with an edge between any two module families if at least one module instance from each family are neighbors in the same gene. Connected components of this graph define the module families to be clustered into a single architecture family. Note that (*B*) uses as input the module families determined by (*A*), but one can use domains as determined by a database search, for example, Pfam domains, if desired. (*C*) In the third stage, architecture scenarios are reconstructed for each architecture family based on a three-step procedure, in which the module trees are reconstructed based on multiple sequence alignments of each module family, these module trees are reconciled to determine ancestral module counts, and the module counts, extant architectures, and known species tree are used to reconstruct the ancestral architectures and ancestral events along each branch.

ous annotations. Thus, we can trace the evolutionary history of clade-specific modules or modules that are not found in current databases (supplementary section 3, Supplementary Material online).

As our goal was to study evolutionary events such as gene merge and split events between multiple species, we excluded any module families that appear in only a single species. Also, as in other works (Fong et al. 2007; Forslund et al. 2008), to mitigate the effects of short length repeat domains and allow for a more efficient algorithm, we collapsed tandem duplicated modules to a single copy and required that a module family appears at most once within an architecture.

Clustering Architectures into Architecture Families

To determine architecture families, we constructed a module adjacency graph, where each vertex represented a module family, and edges were added between two modules if instances of the modules were neighbors within at least one gene. For each connected component within this graph, we identified the set of genes containing at least one module from the cluster and marked them as an architecture family.

From the module adjacency graph, we discovered several highly promiscuous module families that occur in diverse sets of genes. These module families can complicate analysis by creating very large architecture families composed of many distinct gene clusters that share little in common aside from the promiscuous module family. Therefore, we choose to analyze promiscuous module families in a separate analysis (supplementary section 4, Supplementary Material online) and excluded them from our reconstructions. Specifically, module families were removed prior to clustering if they had more than six neighbors; this removed $<0.21\%$ of all modules and $<0.38\%$ of the modules with neighbors.

In addition, to focus on gene fusions and fissions, we filtered our architecture families to those in which one species has a gene with two neighboring modules and another species is either missing one of these modules or has no gene with these modules as neighbors.

Reconstructing Architecture Scenarios

For each architecture family, we reconstructed its evolution by producing an architecture scenario. This is complicated by the fact that inferring architectures in ancestral species implicitly requires inferring module counts. Rather than doing these tasks simultaneously, we adopted a three-stage approach to architecture scenario reconstruction, incorporating known rates of evolutionary events where applicable (supplementary section 5, Supplementary Material online). First, we reconstructed the generation, duplication, and loss history of each module independently of all other modules since these events occur at the module level. Then, we then used these reconstructed module phylogenies to determine ancestral module counts, and finally, we incorporated merge and split events when inferring module groupings into architectures.

In the first stage, we incorporated known rates of evolutionary events to reconstruct the phylogenies of each module family to produce "module trees." This was done by taking the peptide sequences of each module family, aligning them with the MUSCLE software package (Edgar 2004), then reverse translating the result into a (codon-aligned) nucleotide alignment. Module trees were then reconstructed from each nucleotide alignment using the SPIMAP program (Rasmussen and Kellis 2011) configured with model parameters previously determined for the *Drosophila* clade (Rasmussen and Kellis 2011), 100 prescreen iterations, and 50 iterations.

In the second stage, we split modules trees into subtrees containing only descendants of a single common ancestor within or after the root of the species tree (i.e., proper module families). This was achieved by reconciling each module tree to the species tree using maximum parsimonious reconciliation (MPR) (Page 1994; Zmasek and Eddy 2001) and then removing any duplication nodes predating the species tree root (preroot duplications). Each resulting subtree was then rerooted and reconciled repeatedly using MPR until all preroot duplications were removed.

In the third stage, we reconstructed architecture scenarios for each architecture family by combining all of its module trees. From the previous steps of the pipeline, we can infer the extant architectures present at the leaves of the species tree, and we can use the reconciled module trees to infer the ancestral module copy numbers. What remains to be reconstructed is how the ancestral modules combine to form ancestral architectures and what events are responsible for their evolution.

We achieved this reconstruction using a novel maximum parsimony method called STAR-MP (supplementary section 2 and fig. S1, Supplementary Material online), which determines the series of events (generation, duplication, loss, merge, and split) with the least total cost that explain the evolution of the given extant architectures. In this work, we used equal costs for each event, therefore minimizing the total number of events in the reconstruction. Analysis of a subset of families showed that reconstructions are robust to these costs (supplementary section 6, Supplementary Material online).

STAR-MP is a dynamic programming algorithm that first works recursively up the tree to determine the cost of assigning architectures at each node, then works recursively down the tree to assign the most parsimonious architecture at each node as well as the responsible events. In the forward phase, we performed a postorder traversal of the species tree, generating a set of possible architectures for each node by finding all partitions of the available modules, then pruning the resulting list heuristically. For each possible architecture generated, we determined the operations (generation, duplication, loss, merge, split) necessary to transform it into architectures present at the child nodes. Dynamic programming was then used to find the minimum cost-to-go (e.g., minimum total cost along all descendant branches) of assigning the parent architecture. This was repeated until the root of the species tree was reached, at which point the minimum cost architecture was assigned to the root. In the backward phase, we backtracked down the tree to determine the most parsimonious architectures and events at all the internal nodes and edges, respectively. As the maximum parsimonious reconstruction may not be unique, ties were broken randomly to arrive at a single reconstruction.

To measure uncertainty in our reconstructions, we implemented a bootstrapping procedure for STAR-MP. Each module family had 100 module trees reconstructed using SPIMAP on 100 resampled nucleotide alignments. From this set, modules trees were sampled with replacement to be reconciled and analyzed by STAR-MP 100 times, thus generating 100 bootstrapped architecture scenarios.

Validation

Input Validation

A significant challenge of reconstructing architecture evolution is dealing with errors in extant genomes, for example, resulting from sequencing, assembly, or gene model prediction. For example, erroneously connected exons in a gene model or failure to collapse multiple genes into a single gene may cause homologous modules to appear as a single gene in some species but as multiple genes in others. To validate our sequence input, we searched for errors due to gene model or assembly problems. In this section, we provide error rates based on sequence comparison or external evidence; later, in our analysis of architecture scenarios, we will show that these errors have little effect on our biological findings.

In an assembly error, a gene may be separated into multiple scaffolds, or duplicate copies of genes may appear due to undercollapsed scaffolds. In the former case, we would expect a large number of fusion/fission genes to be at the ends of scaffolds. We found that 36.2% (1,486) of the merge/split families to have at least one gene at the end of its scaffold; however, this large percentage is partly attributable to the presence of several short scaffolds in the sequenced genomes. As an alternative measure, 6.51% (2,947) of genes in merge/split families are at the ends of scaffolds compared with 4.85% (6,592) overall, meaning that we possibly find inflated counts for the number of merges and splits. In the latter case, we would expect nearly 100% identity in the sequences. Analysis of the sequences using gene spans with 2,000 base pairs added upstream and downstream reveals 7.31% (300) of the merge/split families have possibly undercollapsed scaffolds (scaffolds contain undercollapsed genes with $\geqslant$98% identity, supplementary section 7.1, Supplementary Material online). Using our rearrangement model, we believe that such families mainly result in double counting of duplications and losses, with little to no effect on the number of merges or splits.

To check for errors due to faulty gene models, we looked at expressed sequence tag (EST) and mRNA-seq evidence for all pairs of neighboring genes (table 1 and supplementary section 7.2, Supplementary Material online). We found that only 0.92% (0.52%) of EST (mRNA-seq) supported neighboring gene models also had an EST (mRNA-seq) spanning

Table 1. EST and mRNA-seq Evidence in Nine *Drosophila* Genomes.

Species	Number of Genes	Number of Genes with EST (mRNA-seq)	Number of Gene Pairs[a]	Number of Gene Pairs with EST (mRNA-seq)[b]	Number of Gene Pairs with Spanning EST (mRNA-seq)[c]	Error Rate (%) of EST (mRNA-seq)[d]
dmel	14,080	12,640 (12,673)	14,052	11,645 (11,895)	78 (35)	0.67 (0.29)
dyak	16,077	1,618	15,335	222	4	1.80
dere	15,044	4,459	14,780	1,556	13	0.84
dana	15,069	5,022	14,680	1,864	24	1.29
dpse	16,099	2,851 (13,721)	15,156	699 (12,092)	13 (70)	1.86 (0.58)
dwil	15,512	4,699	14,442	1,792	17	0.95
dmoj	14,594	4,910 (13,035)	14,209	1,903 (12,123)	19 (82)	1.00 (0.68)
dvir	14,491	5,042	14,216	2,052	23	1.12
dgri	14,982	5,196	13,794	2,133	18	0.84
Total	135,948	46,437 (39,429)	175,882	28,376 (36,110)	262 (187)	0.92 (0.52)

[a]Two adjacent genes on the same strand.
[b]Number of adjacent gene pairs in which both genes have EST (mRNA-seq) evidence.
[c]Number of adjacent gene pairs in which both genes have EST (mRNA-seq) evidence and there exists at least one EST (mRNA-seq) that spans both genes.
[d]Number of gene pairs with spanning EST (mRNA-seq) evidence over the number of gene pairs with EST (mRNA-seq) evidence.

both neighbors, suggesting a low rate of introns misannotated as intergenic modules. The lowest intron annotation error rate was in the well-annotated *D. melanogaster* genome. Larger error rates (e.g., total error rate = 11.53% [EST], 6.66% [mRNA-seq]) occur if we restrict the genes to only those that appear in architecture families (supplementary table S1, Supplementary Material online), but this is likely attributable to the low number of EST (mRNA-seq) supported neighboring gene models in this set. Finally, note that ESTs (mRNA-seqs) only allow us to find introns misannotated as intergenic modules, for example, spurious gene breaks, not intergenic modules misannotated as introns, for example, missed gene breaks.

Methods Validation

Most methods within our phylogenomic pipeline (e.g., residue correlation matrix, OrthoMCL, SPIMAP) have been evaluated in their respective works (Enright et al. 2002; Heger and Holm 2003; Rasmussen and Kellis 2011). To evaluate the last step in this pipeline, our architecture scenario reconstruction algorithm STAR-MP, we simulated module evolution, where simulation parameters were inferred using the maximum parsimony (MP) architecture scenarios reconstructed from real data. Note that this reliance on MP reconstructions means that our simulations underestimate the empirical (and estimated true) event rates.

We started all simulations at the root of the species tree (as was the case for >82.6% of all MP trees) and for each simulation, generated a root architecture, where the number of module families, the number of modules per module family, and the number of connected modules were simulation parameters. To determine the events along each branch, we assumed a separate geometric distribution for each event type (generation, duplication, loss, merge, split) and each branch. The number and type of events along each branch were sampled from these geometric distributions, and an event was applied uniformly among the available modules (generation/duplication/loss), edges (split), or architectures (merge) and was discarded if it was impossible with the given starting architecture. Despite discarding events, event rates for the simulations were similar to the input rates (<6% error).

Using rates estimated from the reconstructed architecture scenarios in *Drosophila*, we simulated 1,000 architecture scenarios and found that STAR-MP has ⩾63.4% sensitivity and ⩾77.8% precision (fig. 4). As in the actual pipeline, the ancestral counts for each module and the architectures at the extant species were provided as input to STAR-MP, accounting for the 100% precision in generation, duplication, and loss events. Evaluation at increased event rates reveals a decrease in sensitivity consistent with a conservative MP algorithm, whereas precision degrades only slightly (supplementary section 7.3 and fig. S2, Supplementary Material online).

Experimental Validation

We investigated transcript evidence (EST and mRNA-seq) at the event and family level, characterizing each event or scenario as "consistent" if there exists no conflicting evidence, "inconsistent" if there exists conflicting evidence, or "unknown" if there exists no evidence (supplementary section 7.4, Supplementary Material online).

We found that 15.1–16.0% of scenarios are consistent and 1.1–1.2% inconsistent, and 23.2–40.9% of merge and split events are consistent and 0.6–1.1% inconsistent

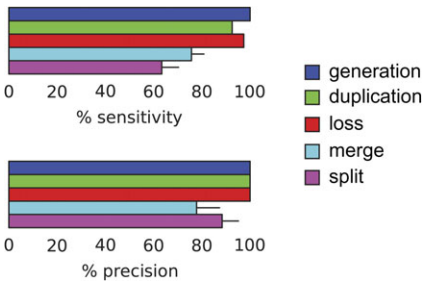

Fig. 4. Reconstruction accuracy of STAR-MP on simulated data sets. Event inference using STAR-MP is both sensitive and precise. Error bars show performance loss due to ties in the MP reconstruction, for example, the MP architecture scenario and the true architecture scenario have equal costs, so events may be missed or extra events may be called in the MP reconstruction.

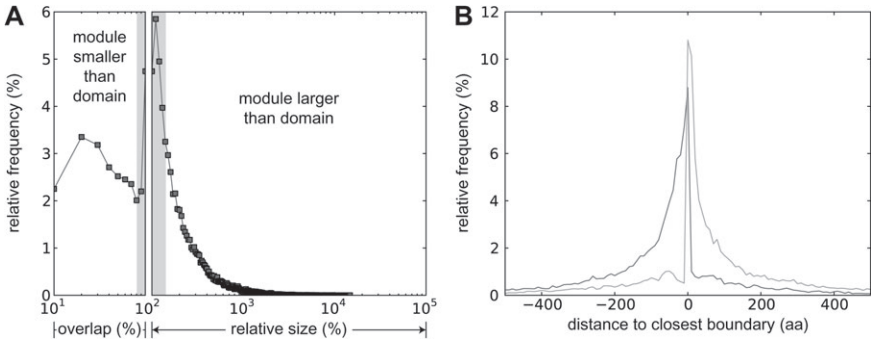

FIG. 5. Correlation of module and domain boundaries. (*A*) For each module, either the overlap (# aa present in both module and domain/domain length) for modules incompletely covered by domains or the relative size (module length/domain length) for modules completely covered by domains was found. 75.6% of modules are equal to or larger than their corresponding domains (relative size $\geqslant$100%), and 28.4% of modules are of similar size to their corresponding domain (overlap $\geqslant$75% or relative size $\leqslant$150%, in gray). Bin size = 10%. (*B*) For each module boundary, the distance to the closest domain boundary was found, where distance = module boundary−domain boundary, blue represents left module boundaries and green represents right module boundaries. Thus, a negative distance in blue and a positive distance in green denote that the module boundary extends further than the domain boundary. Module boundaries tend to be close to domain boundaries or extend further than the closest domain boundary. Bin size = 10 aa.

(supplementary table S2, Supplementary Material online). Although this does not conclusively prove that the merges and splits occur, it does suggest that our reconstructed scenarios and events are not a byproduct of poor gene models.

Results

Using our pipeline, we found 22,813 module families combining in 14,418 architecture families, with 70.4% (10,144) of these architecture families containing only a single module and 28.5% (4,107) containing a merge or split. (All module and architecture families are available online.) The large proportion of single-module families despite such a general definition of gene and module evolution is a testament to the high specificity of our approach. The 4,107 "merge/split" families consist of 12,324 module families covering 45,282 genes and involve at least one gene from 35.9% (4,457/12,431) of FlyBase gene families.

Architecture scenarios were reconstructed for 3,882 families (with 10,448 module families covering 39,476 genes), of which 2,818 (72.6%) had unique maximum parsimony reconstructions; the remaining 5.5% of families had many module families per architecture family and/or large ancestral counts from SPIMAP and were too complicated for MP reconstruction. Mean runtime of STAR-MP was 2.37 s with no bootstrapping and 14.40 s for 100 bootstraps. Analysis of architecture scenarios (see Common Trends in Architecture Scenarios Revealed by STAR-MP Reconstruction and Genome Annotation Errors Contribute to Lineage-Specific Events in Reconstruction) considered nonbootstrapped reconstructions. Reconstructed scenarios typically had high bootstrap support, with a majority (63.2%) of scenarios having a single reconstruction, for example, 100% support on all ancestral architectures and events. Futhermore, each event count had a low standard deviation relative to its mean ($<$0.035), thus demonstrating the robustness of our reconstruction methodology.

Module Boundaries are Driven by Selection: Comparison with Domains and Exons

As our method for finding modules depended solely on sequence similarity rather than relying on previously known structural or functional domain or exon boundaries, we used these two external lines of evidence to study how modules are formed.

Using the curated Pfam-A (version 23.0) (Bateman et al. 2002) domain definitions as a reference, we found that our module detection algorithm tends to avoid over-fragmentation (fig. 5), consistent with the idea of supradomains (Vogel et al. 2004). Furthermore, many modules and domains are also similar in size, and many module boundaries are close to domain boundaries. Note that the long tail in figure 5*A* indicates possible under-fragmentation of domains, which is expected to occur as multiple consecutive domains may have evolved jointly within the ~60 My *Drosophila* clade and thus have been collapsed into a single module.

Comparison between modules and exons reveals similar trends (supplementary fig. S3, Supplementary Material online), with many cases of single module–single exon or single module–multiple exons, and a large percentage (33–42%) of modules lying precisely at an exon boundary (peak at zero distance in supplementary fig. S3A, Supplementary Material online). To study this effect further, we looked at the number of exon-bordering modules (supplementary table S3, Supplementary Material online) and at intron–phase correlations (supplementary table S4, Supplementary Material online). We defined an exon-bordering module as a module in which both boundaries are within $\pm$10 residues of an exon boundary. The unusually high number of exon-bordering modules (observed = 100,974; expected = 2,138; fold = 47.23; $P < 2.23\times10^{-308}$, χ^2 test) indicates exon shuffling as a prominent mechanism of module rearrangement.

Exon shuffling is also supported by a high presence of symmetrical intron phases. An intron has phase zero if it falls between two codons, phase one if it falls after the first

nucleotide within a codon, phase two if it falls after the second nucleotide within a codon, and a module is labeled with the phases of its flanking introns. The splice frame rule (Patthy 1987) states that the phases of introns flanking modules tend to match, as this prevents frameshift mutations after exon shuffling events. Similar to previous analyses (Kaessmann et al. 2002; Liu and Grigoriev 2004; Lee 2009), we found that symmetrical intron phases are enriched ($O = 83{,}394$; $E = 35{,}003$; fold = 2.38; $P < 2.23 \times 10^{-308}$, χ^2 test) and nonsymmetrical intron phases are depleted ($O = 17{,}580$; $E = 65{,}971$; fold = 0.27; $P < 2.23 \times 10^{-308}$, χ^2 test). Furthermore, most of the enrichment in symmetrical intron phases is due to the presence of 0–0 modules; we believe that this enrichment reflects a tendency for exons to be reshuffled at the codon level. Interestingly, though similar trends are seen when comparing Pfam domains and exons (supplementary table S5, Supplementary Material online), fold enrichments and depletions are dramatically increased for modules (e.g., fold values: exon-bordering domains = 2.32, symmetrical intron phases = 1.79, nonsymmetrical phases = 0.58, $P < 2.23 \times 10^{-308}$, χ^2 test), and we found an abundance of 0–0 modules and a lack of 1–1 modules compared with previous analyses. These discrepancies are expected, as previous works used domain definitions produced across many genomes, whereas our modules were detected using data only across the nine *Drosophila* genomes. Regardless of whether domains or modules were used, these results suggest that modules (and domains) are produced through the shuffling of exons; here, a mutational mechanism is made apparent through module (domain) detection.

An alternative explanation for the correlation between module and domain boundaries could be their common correlation with exon boundaries. Thus, we tested whether module boundaries are depleted within domains, which would suggest that modules tend to maintain domains as a unit moreso than would be expected by exon distributions. We found that 7.1% (29,096/410,463) of introns are within $\pm$10 residues of any module boundary, whereas within domains, this percentage decreased to 3.0% (4,451/146,205), supporting our expectation that module boundaries respect domain boundaries (fold = 2.33, $P < 2.23 \times 10^{-308}$, hypergeometric test).

Gene Ontology Terms Associated with Rapid Architecture Evolution Reflect Adaptation

In this section, we address whether certain functions are more likely to be involved in merge and split events. After correcting for possible biases (supplementary section 9, Supplementary Material online), we found seven gene ontology (GO) terms to be enriched across families with merge/split events compared with families without merge/split events ($P < 0.001$, hypergeometric test, false discovery rate correction, table 2). Interestingly, all enriched GO terms are biological processes, and almost, all of them are involved in development.

We hypothesize that although gene fusions and fissions are likely deleterious for most genes, in some cases, they may offer an advantage in terms of adaptability. For example, a domain may be a crucial component in several signaling pathways, each of which requires the domain to interact with a different ligand. Rather than generating the same domain multiple times throughout evolution, a species can duplicate the domain and merge it with others that encode different receptors. Such adaptability may be advantageous in signaling and development (Bhattacharyya et al. 2006; Peisajovich et al. 2010), explaining the enriched GO terms in these categories.

For example, we found an architecture scenario involving the TATA-binding protein (TBP) domain, which associates with different transcription factors to initiate transcription from different RNA polymerases. TBP consists of a highly conserved C-terminal core that binds to the TATA box and interacts with transcription factors and regulatory proteins and a variable N-terminal module. A study of TBP genes hypothesized that the N- and C-terminal modules may have evolved independently of each other and fused together (Sumita et al. 1993). Furthermore, TBP is dependent on upstream activators for promoter specificity; however, fusing TBP to a heterologous DNA-binding domain bypasses the need for a transcriptional activation domain, and the recruitment of TBP with an upstream activation domain provides greater flexibility in promoter arrangement (Xiao et al. 1995). Metazoans may have evolved multiple TBPs to accommodate the vast increase in genes and expression during development and cellular differentiation (Rabenstein et al. 1999).

Protein–Protein Interaction Data Sets Suggest Fusion and Fission of Functionally Complementary Genes

It has been shown that modules that merge or split tend to occur in genes with related functions (Enright et al. 1999; Marcotte et al. 1999; Enright and Ouzounis 2001). This is the basis for the Rosetta Stone model for protein–protein interaction, which suggests that given a Rosetta Stone protein with architecture AB, two proteins with architectures A and B are functionally related and more likely to interact. Possible reasons this trend are that the fusion of neighboring genes allows for tighter coregulation (Bornberg-Bauer et al. 2005), or a single function has separated into two related genes in the case of fission. Here, we determine whether this is the case within the *Drosophila* clade. If so, we may be able to propose new functional annotations for genes.

Within *D. melanogaster*, we identified 1,222 gene partners, where a gene partner consists of two genes connected by a Rosetta Stone protein. That is, for each pair of genes, we defined two sets of modules: the first set contains the modules in gene 1 but not in gene 2, and the second set contains the modules in gene 2 but not in gene 1. To be called a "gene partner," at least one pair of modules, one from each set, must be found fused in a gene in another species. After removing the GO annotations biological process, cellular component, and molecular function, we found that 138 gene partners have both genes annotated with GO terms,

Table 2. GO Enrichment for Genes Undergoing Module Rearrangement.

Rank	GO ID	GO Term	k	m	Fold	P value[a]	P value[b]	Q value[c]
1	GO:0009653	Anatomical structure morphogenesis	426	1,100	1.36	1.61×10^{-14}	2.13×10^{-7}	1.08×10^{-4}
2	GO:0048731	System development	499	1,304	1.34	8.02×10^{-16}	2.34×10^{-8}	1.36×10^{-5}
3	GO:0048856	Anatomical structure development	557	1,465	1.34	5.44×10^{-17}	8.18×10^{-9}	5.53×10^{-6}
4	GO:0007275	Multicellular organismal development	588	1,554	1.33	1.97×10^{-17}	3.37×10^{-9}	3.42×10^{-6}
5	GO:0032502	Developmental process	640	1,709	1.32	7.95×10^{-18}	3.03×10^{-9}	3.42×10^{-6}
6	GO:0032501	Multicellular organismal process	711	1,903	1.31	1.34×10^{-19}	4.23×10^{-10}	8.58×10^{-7}
7	GO:0009987	Cellular process	804	2,218	1.27	3.45×10^{-18}	5.56×10^{-9}	4.51×10^{-6}

[a]Computed using the hypergeometric test, which computes the probability of obtaining at least k annotated families for a given GO term among a data set of size n, using a reference data set containing m such annotated families out of N families. Here, $n = 4,107$ and $N = 14,418$.
[b]P values corrected for length bias.
[c]P values corrected for length bias and multiple hypothesis testing (false discovery rate).

and of these, 114 (82.6%) share at least one GO term. By selecting random gene partners (to control for length bias, these partners were selected from the set of 208 genes that form the 1,222 partners), we observed that 61.8% share a GO term on average. This suggests that genes are more likely to have related functions if they have modules that merge or split (fold = 1.34, $P < 0.001$), though the cause and effect may be the reverse.

Common Trends in Architecture Scenarios Revealed by STAR-MP Reconstruction

Our architecture scenarios that involve module merges and splits cover 4,107 architecture families, 12,324 module families, and 45,282 genes. However, many of these families have very simple scenarios. Most (2,295, 55.9%) contain only two modules (fig. 6), and many (1,007, 24.5%) contain one gene in each of the nine species. These single gene families frequently consist of distinct subtrees, one with a single module A and another with merged architecture AB, implying a single generation and merge of module B. The second most frequent scenario (767 families, 18.7%) consists of ten sequences across nine species, corresponding to one fused gene in eight species and two fragmented genes in one species. This suggests that fragmented genes (and as we will see, fused genes) may be lineage specific, an idea we will later revisit.

Using our 3,882 reconstructed architecture scenarios, we studied the distributions of each of our events (table 3, fig. 7, and supplementary fig. S6, Supplementary Material online).

For generation events, we found that most modules (8,339/10,448; or 79.8%) are generated at the species tree root (fig. 7) and were therefore inferred to exist prior to the *Drosophila* speciation. A previous study on the origin of new genes in the *D. melanogaster* species subgroup found that de novo gene origination from noncoding sequences accounts for 11.9% of new genes (Zhou et al. 2008), suggesting that partial gene origination may not be rare (Long et al. 2003).

For duplication and loss events, we observed that losses occur 2.29 times more than duplications, which is consistent with previous studies at the gene level that found factors of 1.78–3.18 (Rasmussen and Kellis 2011). The large number of duplications relative to losses arises due to paraphyletic modules (modules that appear in an ancestor but do not appear in all descendants of that ancestor), which could require multiple loss events, and also due to modules trees that are incongruent with the species tree so that during reconciliation, a single ancient duplication is compensated for with multiple losses.

Lastly, for merge and split events, a comparison of their counts revealed a 0.86:1 merge-to-split ratio, which at first seemed inconsistent with previous studies suggestingthat

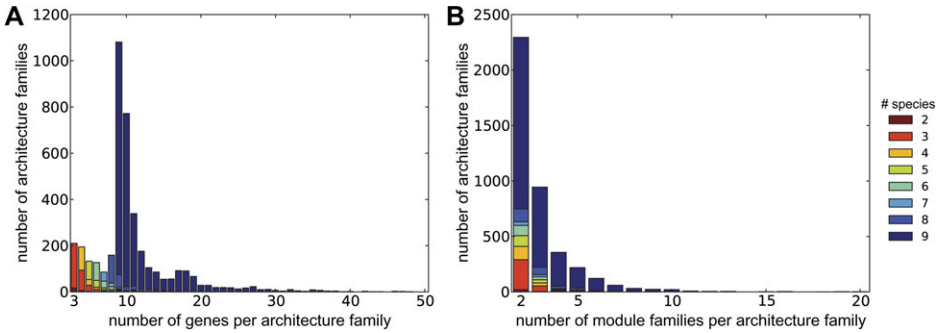

FIG. 6. Distribution of architecture family sizes. (*A*) The number of sequences per architecture family (20 families with more than 50 sequences not shown), and (*B*) the number of module families per architecture family (3 families with more than 20 modules not shown) are shown. Color denotes the number of species represented in the architecture family. Many families have simple evolutionary histories, for example, have a single gene per species or contain only two interacting modules.

Table 3. Inferred Evolutionary Events Across Architecture Scenarios.

Event or Ratio[a]		G	D	L	M	M_s	S	S_s	D/L	M/S	M_s/S_s
Full[b]	**# Number of Events[d]**	2,109	4,302	9,873	4,876	2,952	5,659	559	1:2.29	0.86:1	5.28:1
	# Number of Scenarios[e]	1,520	1,775	2,961	2,242	955	2,880	257	1:1.67	0.78:1	3.71:1
	Percentage of scenarios	39.2	45.7	76.3	57.8	24.6	74.2	6.6			
Conserved[c]	**# Number of Events[d]**	1,279	1,426	5,763	2,567	1,509	2,880	235	1:4.04	0.89:1	6.42:1
	# Number of Scenarios[e]	1,015	940	1,954	1,374	529	1,747	81	1:2.08	0.79:1	6.53:1
	Percentage of scenarios	40.7	37.7	78.4	55.1	21.2	70.1	3.3			

[a]G, generation; D, duplication; L, loss; M, merge; S, split. M_s and S_s represent simple merges and splits, that is, merges unaccompanied by generation or duplication events and splits unaccompanied by duplication or loss events.
[b]Counts aggregated across all 3,882 reconstructed architecture scenarios.
[c]Counts aggregated across a conservative set of 2,506 reconstructed architecture scenarios with limited genome annotation errors.
[d]Total number of events across all architecture scenarios.
[e]Number of architecture scenarios with at least one branch having the event type.

fusion occurs more often than fission by a factor of 2.6–5.6 (Snel et al. 2000; Kummerfeld and Teichmann 2005; Fong et al. 2007). However, one key difference in this analysis is that we measured individual events, as opposed to simply observing the presence of fused and fragmented extant genes, and we measured events over a smaller higher resolution time scale (the 62 My *Drosophila* clade vs. all three domains of life diverging over 3.5 billion years). Furthermore, other studies do not indicate how they handle complicated events such as partial gene duplication (architecture AB to

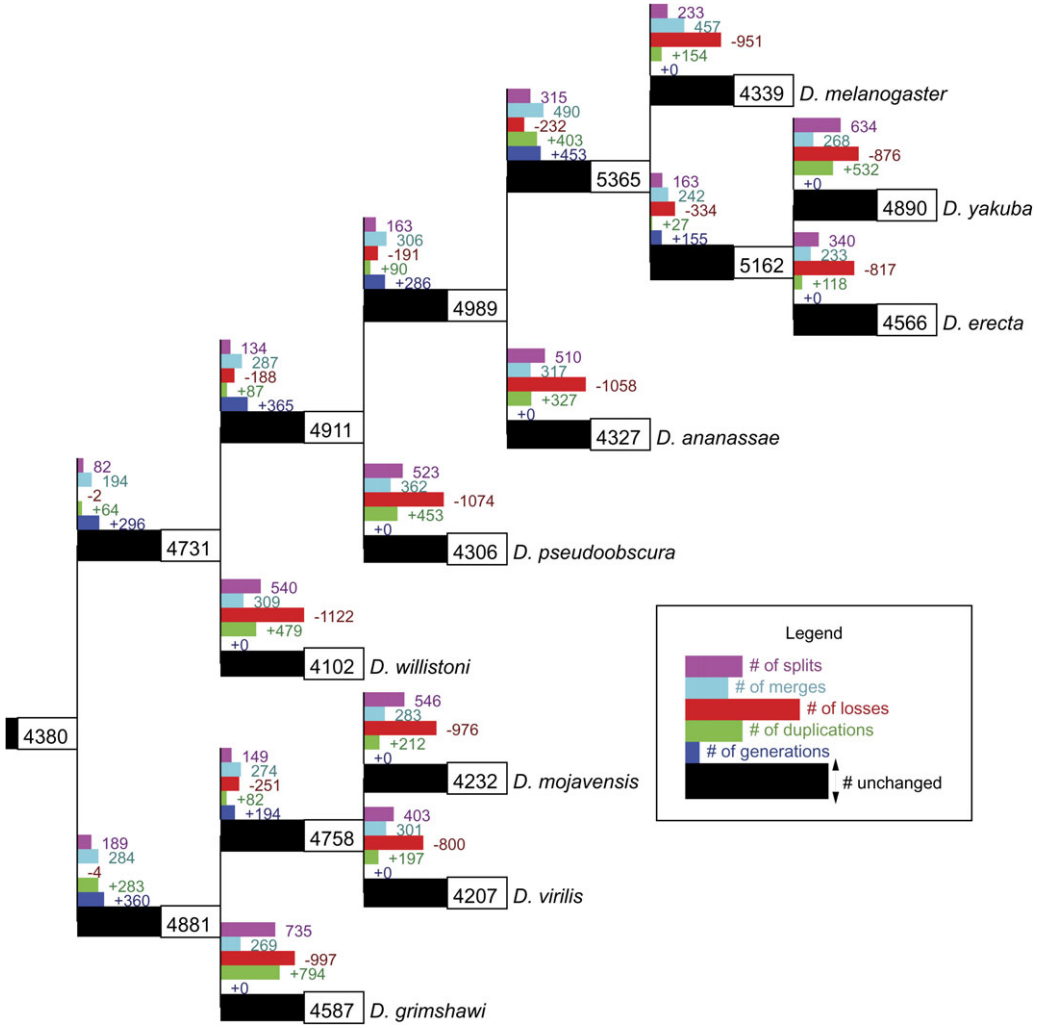

FIG. 7. Total counts of evolutionary events inferred on the nine *Drosophila* phylogeny by STAR-MP. Many evolutionary events are inferred along each branch (counts aggregated across 3,882 architecture scenarios). The large number of losses is consistent with ancient duplications followed by many compensatory losses. Many merges and splits are located along leaf branches, indicating that many fusion and fission genes may be lineage specific. Histograms of event counts are shown along each branch, and the number of modules in a species is displayed at each species node, where counts are totaled across all architecture scenarios.

Table 4. Retainment of Ancestral Architectures by Merge and Split Events.

MERGES	All (%)	Without Generation (%)	With Generation (%)
Number of events[a]	4,876	3,623 (74.3)	1,253 (25.7)
Retained at least one split architecture[b,e]	4,512 (92.5)	3,437 (94.9)	1,075 (85.8)
Retained both split architectures[c,e]	2,688 (55.1)	2,688 (74.2)	n/a
SPLITS	**All (%)**	**Without Loss (%)**	**With Loss (%)**
Number of events[a]	5,659	2,683 (47.4)	2,976 (52.6)
Retained merged architecture[d,e]	1,943 (34.3)	1,844 (68.7)	99 (3.3)

[a]Total number of merge/split events, as well as whether these events are merges with a newly generated module (e.g., A → AB) or splits that also lose an associated split module (e.g., AB → A). Percentages out of the total number of merge/split events.
[b]Number of merges that retain at least one ancestral split architecture (e.g., A,B → A,AB).
[c]Number of merges that retain both ancestral split architectures (e.g., A,B → A,B,AB).
[d]Number of splits that retain the ancestral merged architecture (e.g., AB → AB,A,B).
[e]Percentages out of the number of events in the top row.

architectures AB and A) and partial gene loss (e.g., architecture AB to architecture A). We considered the former example to require a split prior to duplication and the latter to require a split prior to loss, whereas other models may have allowed for the duplication and loss to occur without an accompanying split. Investigation of our reconstructed architecture scenarios showed that many splits are due to such partial duplications and losses; by considering only "simple" merges and splits that are unaccompanied by generation, duplication, or loss events, the merge-to-split ratio became 5.28:1, which is much more comparable to previously determine ratios.

This last observation prompted us to also analyze the co-occurrence of events. The first trend we found is that merge and split events tend to co-occur within module and architecture families. There are 1,264 scenarios (32.6% of all reconstructed scenarios, 25.9% of scenarios with merge events, 22.3% of scenarios of split events) with both merge and split events. Furthermore, 2,419 module families are involved in both merge and split events (42.9% of the 5,645 module families that undergo a merge, 34.3% of the 7,049 module families that undergo a split). This suggests that modules that undergo a merge or split event are more likely to undergo further rearrangement (compared with the 22,861 module families in *Drosophila*, fold = 1.39, hypergeometic test, $P = 1.31 \times 10^{-108}$).

Another interesting relationship is how merge and splits events co-occur with the other events (table 4). For example, most (74.3%) merges occur between existing (nongenerated) modules, and most (92.5%) retain at least one premerge architecture (due to a previous duplication event). This is similar to cases such as *jingwei* where a duplication and merge has preserved the parental gene forms. In contrast, we found that most (52.6%) split events occur with the loss of a resulting split module, and few (34.3%) retain the presplit architecture.

Genome Annotation Errors Contribute to Lineage-Specific Events in Reconstruction

We found that 57.4% of all merge events and 78.9% of all split events occur along a branch leading to an extant species (supplementary table S6, Supplementary Material online). This could suggest that merge and split events tend to be lineage specific, as found in previous studies of *Drosophila* (Zhou et al. 2008; Rogers et al. 2009), or it could be an artifact of our pipeline arising from poor gene models and architecture annotations. For example, the *D. melanogaster* lineage contains 9.4% of all merge events and 16.3% of all lineage-specific merge events even though its branch accounts for only 2.9% of the total branch length within the species tree and 3.7% of the total leaf branch lengths. This genome also accounts for 14.7% (446) of the 3,044 fused genes for which the split form consists of two adjacent genes, compared with an average of 10.7% (295–341) in all other genomes. However, since *D. melanogaster* has the best annotated genome and lowest gene model error rate (table 1), these large percentages could be explained by genes being erroneously called as separate genes in other species and correctly called as a single gene in *D. melanogaster*, leading to a MP reconstruction in which a single merge event has occurred along the *D. melanogaster* branch.

Due to such potential anomalies, we would like a rough estimate of how many architecture families could erroneously contain merge or split events. Though we have previously validated our sequence input, we also decided to consider a highly conservative set of architecture families, which we defined as families in which no genes are neighbors, no genes are at the ends of scaffolds, and no genes have transitive BLAST hits through alternatively spliced forms. This last filter removes possible spurious gene fusions and fissions, in which part of the fused gene is found in an alternative transcript but not in the longest transcript.

Filtering the 4,107 architecture families involving module merges or splits resulted in a set of 2,506 families (61.0% of original set) with 6,120 modules (49.7%) covering 21,780 genes (48.0%). This implies that up to 39.0% of the "merge/split" architecture families could be affected by genome annotation errors or alternative transcripts that were not considered. Within the conservative set, 2,492 families with 6,022 modules covering 21,518 genes had reconstructed architecture scenarios. Note the 2-fold decrease in the number of sequences represented. This is expected, as our conservative set likely discarded many true examples of gene fusion and fission; for example, all scenarios with adjacent genes merging or a gene splitting into two adjacent

genes were removed, despite both of these are being valid potential mechanisms.

This conservative set of families removed 54.7% of lineage-specific merges and 48.8% of lineage-specific splits. However, 49.4% of the remaining (conservative) merge events and 79.3% of the remaining split events are still lineage specific, and the percentage of merge events in the *D. melanogaster* lineage was only reduced from 9.4% to 6.8% (percentage of lineage-specific merge events reduced from 16.3% to 13.7%) (supplementary table S9, Supplementary Material online), suggesting that lineage-specific events are not solely a byproduct of poor gene annotations.

Considering all architecture families, the conservative filter retained 12,408 families (86.1% of original set) with 16,178 modules (70.9%) covering 84,496 genes (75.3%). Though ratios and folds changed, all results within the previous sections hold (GO enrichment: supplementary table S7, Supplementary Material online; PPI: supplementary section 11, Supplementary Material online; event counts: table 3, supplementary tables S8–S9 and fig. S7, Supplementary Material online).

Phylogenomic Pipeline Recovers Previously Known Examples of Chimeric Genes

Zhou et al. (2008) and Rogers et al. (2009) previously identified 47 unique chimeric genes in *D. melanogaster*, 21 of which were also identified by our algorithm (supplementary table S10, Supplementary Material online), yielding a sensitivity of 44.7%. However, Zhou et al. (2008) allowed chimeric genes to arise from a single parental sequence recruiting sequences from other intronic or intergenic sequences or from repetitive elements; this resulted in 32 of their chimeric genes having a single parental gene. Such chimeric genes might not have been detected by our pipeline since a gene subsequence must have had a hit for it to propagate through our module detection algorithm, and our use of protein sequences eliminated any possible hits to intronic or integenic sequences. By considering only chimeric genes that have two or more parental genes, our sensitivity rises to 60% (9/15). The remaining chimeric genes were not identified due to no hits found (one), no hits found satisfying the percent identity threshold (one), frameshift mutations (one), overlapping alignments (two), or underclustering of modules into module families (one). The first two reasons are a consequence of the BLAST step in our pipeline, where we chose thresholds consistent with previous studies in phylogenomics (Rasmussen and Kellis 2007). Similarly, regarding the last reason, we chose a clustering threshold for OrthoMCL consistent with previous studies (Enright et al. 2002).

Both Zhou et al. (2008) and Rogers et al. (2009) used BLASTn to compare coding sequences, and they used different filters, for example, they kept only the top hits or used different alignment length and percent identity thresholds. In our pipeline, we used peptide sequences and BLASTp to compare sequences in our pipeline as peptide homology is more sensitive than nucleotide homology. However, our choice to use BLASTp also eliminated our ability to detect frameshift mutations. Investigation of nucleotide alignments suggests that frameshift mutations account for a small percentage (~0.58%) of total alignments and would increase the number of genes participating in merge/split families by <3.15% (supplementary section 12, Supplementary Material online). Future investigation may incorporate these alignments into our pipeline.

Both cases of overlapping alignments had nearly full overlaps among the three sequences, indicating that the three genes were likely duplicate copies rather than two parental sequences and one chimeric child. Aside from sequence changes in the data sets that could have caused nonoverlapping alignments to now appear as overlapping, remember that we also extended our alignments using LALIGN, whereas Zhou et al. (2008) and Rogers et al. (2009) used BLASTn alignments. Manual inspection of the alignments suggests that the full overlapping alignments are correct, and the two cases correspond to nonchimeric genes.

Gene Fusion and Fission Events Reflect a Small Number of Common Mechanisms

In this section, we consider possible mechanisms for generating new architectures that require merges and/or splits (fig. 8), show a concrete example of the mechanism, and determine how often each mechanism occurs within *Drosophila* (supplementary section 13, Supplementary Material online, catalog of genes by mechanism available online).

The first mechanism allows neighboring genes to merge or split, which could occur by mutations that alter start and stop codons. Allowing for the duplication of genes or subsequences before merges or splits, we found that 1,681 modules and 6,713 genes (16.4% and 17.2% of the modules/genes participating in merge/split events) possibly undergo this mechanism. Of course, such merges and splits are also the most suspect, as they could be caused by poor gene calls. Looking to EST (mRNA-seq) evidence, we found 274 (236) of the above genes are inconsistent with ESTs (mRNA-seqs), 5,863 (4,534) genes have no ESTs (mRNA-seqs), and 576 (1,943) genes are consistent with ESTs (mRNA-seqs). Other more complicated mechanisms may also explain these fusions and fissions. For example, a merged gene that is found between the ancestral split genes (not necessarily as neighbors, example in supplementary fig. S8, Supplementary Material online) may be the result of large loop mismatch repair or replication slippage (Rogers et al. 2009). We found that 32 modules and 19 genes (0.3% and 0.05%) possibly result from these mechanisms.

The second mechanism was introduced with the case of *jingwei* (supplementary fig. S9, Supplementary Material online), an example which is recovered by our pipeline. Here, a retrotransposed copy of a gene is inserted into another gene and exons are combined to produce a new gene (though a fusion of the transcripts followed by retrotransposition is also possible; Akiva et al. 2006). Such an event would correspond to a duplication and merge in our algorithm, but duplications and splits are also possible if a partial retrotransposition occurs. We found that 1,904

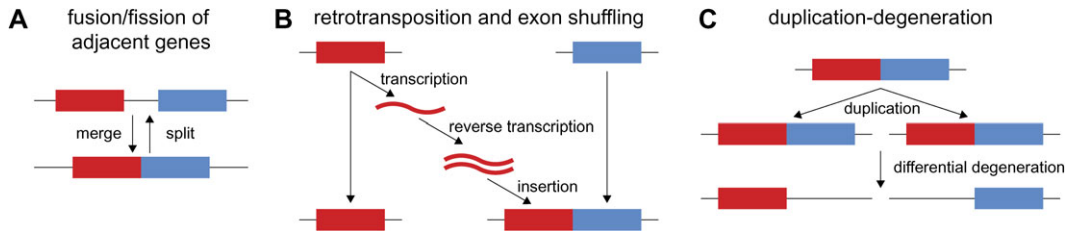

FIG. 8. Mechanisms for generating fused and fragmented architectures. (*A*) Two adjacent genes merge into a single gene, or a single gene splits into two genes. (*B*) A retrotransposed copy of a gene combines with exons from another gene. (*C*) A chromosomal segment duplicates, and alternative portions of the duplicates are lost.

modules and 2,023 genes (18.5% and 5.18% of modules/genes participating in merge/split events) potentially result from this mechanism. In comparison, previous studies found that retrotransposition accounts for 12.2% of chimeric genes in *D. melanogaster* (Zhou et al. 2008).

The third mechanism involves segmental duplication followed by differential loss and was observed in the monkey king family (Wang et al. 2004). Though we did not find this example in our data set as the events occur in a sister group of *D. melanogaster* not included in our nine species, we found that 60 modules and 79 genes (0.6% and 0.2% of the modules/genes participating in merge/split events) result from this mechanism. An example is the evolution of the *rhea* family (fig. 9).

Discussion

We have presented a novel model of evolution that captures module-level events such as generation, duplication, loss, merge, and split, all of which lead to new module architectures, and we have also introduced a maximum parsimony algorithm STAR-MP for tracing architecture evolution and demonstrated its accuracy in simulation. Furthermore, using our architecture-aware phylogenomic pipeline on a clade of nine *Drosophila* species, we have provided the most complete picture yet of gene and module evolution in a complete genome across multiple species.

Unlike conventional gene tree reconstruction methods, our approach incorporated module architectures and was thus able to model how genes across gene families may be related, as indicated by the presence of similar modules or architectures. Also, unlike most architecture-aware phylogenomic analyses, our approach found gene modules de novo rather than relying on external domain models, and our reconstruction pipeline traced gene evolution while incorporating sequence information and providing statistical bootstrapping support.

Our results revealed that merges are more prevalent than splits as reported in previous directed studies. We also showed that merge and split events tend to occur more frequently when duplications have also retained the original architectures, likely allowing new functions to be generated by the newly formed merged or split gene while retaining the original functions of the ancestral genes. Our approach should enable the systematic study of whether gene merges and splits are enriched in alternatively spliced genes, and how often an alternatively spliced form carries the original architecture. We did not focus on this question here, as we only used a single splice form (the longest polypeptide) for each gene in this initial study and because alternative splice forms are only well annotated in *D. melanogaster* and not across the *Drosophila* clade.

In our study, we used SPIMAP for phylogenetic reconstruction of module trees; SPIMAP is a species tree-aware program that can maximize phylogenetic accuracy for small sequences. This is especially important as phylogenetic accuracy is dependent on the length of the sequences compared, which can make subgene-level phylogenetic reconstruction (as in our module trees) especially error-prone in absence of a known species tree.

We used equal event costs and ignored branch lengths (both within the known species tree and the reconstructed module phylogenies) when reconstructing architecture scenarios with STAR-MP. This assumes that events are equally likely across all branches regardless of event type or branch lengths. Although we could have incorporated the inferred merge-to-split ratios (as reported in Common Trends in Architecture Scenarios Revealed by STAR-MP Reconstruction) when assigning event costs, we wished to avoid such circular dependencies. Future studies may investigate ways to estimate these rates and incorporate them in a probabilistic or weighted parsimony framework.

A major bottleneck of architecture reconstruction algorithms is the enumeration of possible architectures, which can use both the order of modules within architectures and the number of architecture instances within families; thus, the number of possible parent architectures given two children architectures can be intractably large. STAR-MP relied on heuristics to limit the set of parent architectures for increased efficiency, and using a maximum parsimony approach, it was possible to consider a large number of parent architectures since computing the rearrangement cost for each combination of parent and children architectures is relatively fast. However, future work may require a better understanding of architecture rearrangements to better sample the full architecture space. Further analysis, for example looking at how often modules change order, may provide insight into architecture arrangements and help us develop a more biologically relevant model. Similarly, we can examine whether more complicated events such as module inversion are required for accurate architecture reconstruction.

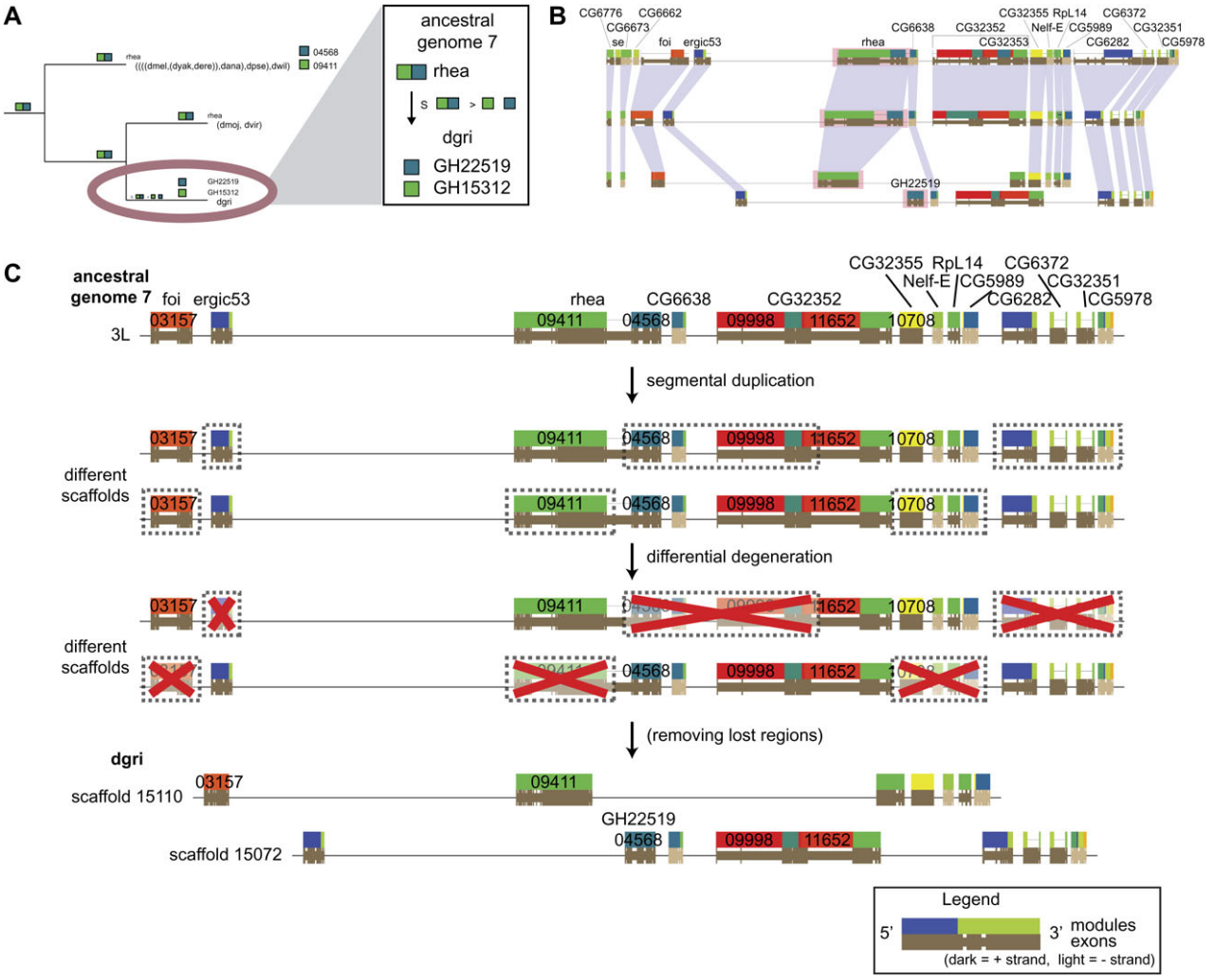

FIG. 9. The inferred evolutionary history of *GH22519* in *D. grimshawi* through duplication–degeneration of *rhea*. (*A*) The MP architecture scenario. (The full MP architecture scenario is available for download.) Most species have the module 09411 and 04568 fused in a single gene *rhea*. However, *dgri* has the two modules in separate genes, with the *rhea* ortholog containing module 09411 and the *GH22519* gene containing module 04568. The MP reconstruction infers a split along the branch leading to *dgri*. Note that in the full MP architecture scenario, there is a second gene with module 09411 in the (*dmel*,(*dyak*,*dere*)) ancestor, which is caused by the module tree (incorrectly) grouping *dmel* and *dere* together. This results in likely spurious duplication, loss, and split events being inferred within the *melanogaster* subgroup. (*B*) A genome level view shows that *rhea* and *GH22519* in *dgri* are found on two scaffolds that alternately contain orthologs to the other eight genomes. (*C*) The inferred evolutionary history of *rhea* and *GH22519* in *dgri* through segmental duplication followed by differential degeneration. Instead of losing the entire *rhea* gene in one of the duplicates, *rhea* undergoes alternative module loss, with each copy retaining one module of the original *rhea* gene. This results in two genes that appear fused in the other species and fragmented in *dgri*.

The methods presented here relied on parsimonious reconstructions of evolutionary histories, which allowed us to limit the number of scenarios to consider, resulting in high speed and accuracy. A major challenge going forward is to extend these methods to propagate sequence information across all possible reconstructions, similar to existing Bayesian and maximum likelihood phylogenetic methods, which we believe could better capture the evolutionary history of architecture families. In particular, such probabilistic methods could allow for the modeling of branch lengths within an architecture DAG (rather than being limited to architecture scenarios) and thus place evolutionary events at specific timepoints within the species history. This could also allow the simultaneous modeling of both sequence and architecture evolution, rather than the current approaches of utilizing sequence to reconstruct module trees and then either using architecture to reconstruct architecture scenarios or using reconciliation to determine module insertions and deletions.

Finally, although we have only focused on the *Drosophila* clade, increasing numbers of complete genomes are becoming commonplace across vertebrates and fungi, especially in mammals and yeast species. Further analysis of such genomes using our methods can reveal many new insights into module neofunctionalization and the emergence of new gene functions through module-level events.

Supplementary Material

Supplementary sections 1–13, tables S1–S10, and figures S1–S9 are available at *Molecular Biology and Evolution* online (http://www.mbe.oxfordjournals.org/).

Acknowledgments

We thank the MIT CompBio group for helpful comments, feedback, and discussions, and modENCODE for early release of their RNA-seq data. This material is based upon work supported under a National Science Foundation (NSF) Graduate Research Fellowship to Y.W. and NSF CAREER award 0644282 to M.K.

References

Akiva P, Toporik A, Edelheit S, Peretz Y, Diber A, Shemesh R, Novik A, Sorek R. 2006. Transcription-mediated gene fusion in the human genome. *Genome Res*. 16:30–36.

Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 25:3389–3402.

Apic G, Gough J, Teichmann SA. 2001. An insight into domain combinations. *Bioinformatics* 17:S83–S89.

Apic G, Huber W, Teichmann SA. 2003. Multi-domain protein families and domain pairs: comparison with known structures and a random model of domain recombination. *J Struct Funct Genomics* 4:67–78.

Arvestad L, Lagergren J, Sennblad B. 2009. The gene evolution model and computing its associated probabilities. *JACM* 56:1–44.

Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy S, Griffiths-Jones S, Howe KL, Marshall M, Sonnhammer ELL. 2002. The Pfam protein families database. *Nucleic Acids Res*. 30:276–280.

Behzadi B, Vingron M. 2006. Reconstructing domain compositions of ancestral multi-domain proteins. In: Bourque G, El-Mabrouk N, editors. Comparative genomics. Lecture Notes in Computer Science. Vol. 4205. Berlin-Heidelberg (Germany): Springer. p. 1–10.

Bhattacharyya RP, Reményi A, Yeh BJ, Lim WA. 2006. Domains, motifs, and scaffolds: the role of modular interactions in the evolution and wiring of cell signaling circuits. *Annu Rev Biochem*. 75:655–680.

Bornberg-Bauer E, Beaussart F, Kummerfeld SK, Teichmann SA, Weiner J. 2005. The evolution of domain arrangements in proteins and interaction networks. *Cell Mol Life Sci*. 62:435–445.

Butler G, Rasmussen MD, Lin MF, et al. (51 co-authors). 2009. Evolution of pathogenicity and sexual reproduction in eight Candida genomes. *Nature* 459:657–662.

Chen K, Durand D, Farach-Colton M. 2000. NOTUNG: a program for dating gene duplications and optimizing gene family trees. *J Comput Biol*. 7:429–447.

Courseaux A, Nahon JL. 2001. Birth of two chimeric genes in the hominidae lineage. *Science* 291:1293–1297.

David LA, Alm EJ. 2011. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature* 469:93–96.

Durrens P, Nikolski M, Sherman D. 2008. Fusion and fission of genes define a metric between fungal genomes. *PLoS Comput Biol*. 4:e1000200.

Edgar R. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. 32:1792–1797.

Eisen JA. 1998. Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Res*. 8:163–167.

Eisen JA, Fraser CM. 2003. Phylogenomics: intersection of evolution and genomics. *Science* 300:1706–1707.

Enright A, Ouzounis C. 2001. Functional associations of proteins in entire genomes by means of exhaustive detection of gene fusions. *Genome Biol*. 2:research0034.1–research0034.7.

Enright AJ, Iliopoulos I, Kyrpides NC, Ouzounis CA. 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402:86–90.

Enright AJ, Van Dongen S, Ouzounis CA. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*. 30:1575–1584.

Felsenstein J. 1981. Evolutionary trees from dna sequences: a maximum likelihood approach. *J Mol Evol*. 17:368–376.

Fong JH, Geer LY, Panchenko AR, Bryant SH. 2007. Modeling the evolution of protein domain architectures using maximum parsimony. *J Mol Biol*. 366:307–315.

Forslund K, Henricson A, Hollich V, Sonnhammer ELL. 2008. Domain tree-based analysis of protein architecture evolution. *Mol Biol Evol*. 25:254–264.

Goodman M, Czelusniak J, Moore G, Romero-Herrera A, Matsuda G. 1979. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*. 28:132–163.

Gough J. 2005. Convergent evolution of domain architectures (is rare). *Bioinformatics* 21:1464–1471.

Hahn MW, De Bie T, Stajich JE, Nguyen C, Cristianini N. 2005. Estimating the tempo and mode of gene family evolution from comparative genomic data. *Genome Res*. 15:1153–1160.

Hahn MW, Demuth JP, Han SG. 2007. Accelerated rate of gene gain and loss in primates. *Genetics* 177:1941–1949.

Hahn MW, Han MV, Han SG. 2007. Gene family evolution across 12 *Drosophila* genomes. *PLoS Genet*. 3:e197.

Han JDJ, Dupuy D, Bertin N, Cusick ME, Vidal M. 2005. Effect of sampling on topology predictions of protein-protein interaction networks. *Nat Biotechnol*. 23:839–844.

Heger A, Holm L. 2003. Exhaustive enumeration of protein domain families. *J Mol Biol*. 328:749–767.

Huang X, Miller W. 1991. A time-efficient, linear-space local similarity algorithm. *Adv Appl Math*. 12:337–357.

Hunter S, Apweiler R, Attwood TK, et al. (38 co-authors). 2009. InterPro: the integrative protein signature database. *Nucleic Acids Res*. 37:D211–D215.

Jones CD, Begun DJ. 2005. Parallel evolution of chimeric fusion genes. *Proc Natl Acad Sci U S A*. 102:11373–11378.

Jones CD, Custer AW, Begun DJ. 2005. Origin and evolution of a chimeric fusion gene in drosophila subobscura, D. madeirensis and D. guanche. *Genetics* 170:207–219.

Kaessmann H, Zöllner S, Nekrutenko A, Li WH. 2002. Signatures of domain shuffling in the human genome. *Genome Res*. 12:1642–1650.

Kummerfeld SK, Teichmann SA. 2005. Relative rates of gene fusion and fission in multi-domain proteins. *Trends Genet*. 21:25–30.

Lee B. 2009. Comparison of exon-boundary old and young domains during metazoan evolution. *Genomics Inform*. 7:131–135.

Liu M, Grigoriev A. 2004. Protein domains correlate strongly with exons in multiple eukaryotic genomes—evidence of exon shuffling? *Trends Genet*. 20:399–403.

Long M, Betran E, Thornton K, Wang W. 2003. The origin of new genes: glimpses from the young and old. *Nat Rev Genet*. 4:865–875.

Long M, Langley CH. 1993. Natural selection and the origin of jingwei, a chimeric processed functional gene in Drosophila. *Science* 260:91–95.

Long M, Rosenberg C, Gilbert W. 1995. Intron phase correlations and the evolution of the intron/exon structure of genes. *Proc Natl Acad Sci U S A*. 92:12495–12499.

Long M, Wang W, Zhang J. 1999. Origin of new genes and source for N-terminal domain of the chimerical gene, jingwei, in Drosophila. *Gene* 238:135–141.

Lynch M, Conery JS. 2000. The evolutionary fate and consequences of duplicate genes. *Science* 290:1151–1155.

Marchler-Bauer A, Anderson JB, Cherukuri PF, et al. (24 co-authors). 2005. CDD: a conserved domain database for protein classification. *Nucleic Acids Res*. 33:D192–D196.

Marcotte EM, Pellegrini M, Ng HL, Rice DW, Yeates TO, Eisenberg D. 1999. Detecting protein function and protein-protein interactions from genome sequences. *Science* 285:751–753.

Moore AD, Björklund AK, Ekman D, Bornberg-Bauer E, Elofsson A. 2008. Arrangements in the modular evolution of proteins. *Trends Biochem Sci*. 33:444–451.

Murzin AG, Brenner SE, Hubbard T, Chothia C. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*. 247:536–540.

Niimura Y, Nei M. 2007. Extensive gains and losses of olfactory receptor genes in mammalian evolution. *PLoS One* 2:e708.

Ohno S. 1970. Evolution by gene duplication. New York: Springer-Verlag.

Page RDM. 1994. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst Biol*. 43:58–77.

Pasek S, Risler JL, Brézellec P. 2006. Gene fusion/fission is a major contributor to evolution of multi-domain bacterial proteins. *Bioinformatics* 22:1418–1423.

Patthy L. 1987. Intron-dependent evolution: preferred types of exons and introns. *FEBS Lett*. 214:1–7.

Peisajovich SG, Garbarino JE, Wei P, Lim WA. 2010. Rapid diversification of cell signaling phenotypes by modular domain recombination. *Science* 328:368–372.

Ponting CP, Russell RR. 2002. The natural history of protein domains. *Annu Rev Biophys Biomol Struct*. 31:45–71.

Przytycka T, Davis G, Song N, Durand D. 2006. Graph theoretical insights into evolution of multidomain proteins. *J Comput Biol*. 13:351–363.

Rabenstein MD, Zhou S, Lis JT, Tjian R. 1999. Tata box-binding protein (tbp)-related factor 2 (trf2), a third member of the tbp family. *Proc Natl Acad Sci U S A*. 96:4791–4796.

Rasmussen MD, Kellis M. 2007. Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes. *Genome Res*. 17:1932–1942.

Rasmussen MD, Kellis M. 2011. A Bayesian approach for fast and accurate gene tree reconstruction. *Mol Biol Evol*. 28:273–290.

Rogers RL, Bedford T, Hartl DL. 2009. Formation and longevity of chimeric and duplicate genes in Drosophila melanogaster. *Genetics* 181:313–322.

Rogers RL, Bedford T, Lyons AM, Hartl DL. 2010. Adaptive impact of the chimeric gene Quetzalcoatl in Drosophila melanogaster. *Proc Natl Acad Sci U S A*. 107:10943–10948.

Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*. 4:406–425.

Schultz J, Milpetz F, Bork P, Ponting CP. 1998. Smart, a simple modular architecture research tool: Identification of signaling domains. *Proc. Natl. Acad. Sci. U. S. A*. 95:5857–5864.

Shih HJ, Jones CD. 2008. Patterns of amino acid evolution in Drosophila ananassae chimeric gene, siren, parallel those of other Adh-derived chimeras. *Genetics* 180:1261–1263.

Snel B, Bork P, Huynen M. 2000. Genome evolution: gene fusion versus gene fission. *Trends Genet*. 16:9–11.

Song N, Joseph JM, Davis GB, Durand D. 2008. Sequence similarity network reveals common ancestry of multidomain proteins. *PLoS Comput Biol*. 4:e1000063.

Suhre K, Claverie JM. 2004. FusionDB: a database for in-depth analysis of prokaryotic gene fusion events. *Nucleic Acids Res*. 32:D273–D276.

Sumita K, Makino Y, Katoh K, Kishimoto T, Muramatsu M, Mikoshiba K, Tamura T. 1993. Structure of a mammalian TBP (TATA-binding protein) gene: isolation of the mouse TBP genome. *Nucleic Acids Res*. 21:2769.

Tamura K, Subramanian S, Kumar S. 2004. Temporal patterns of fruit fly (Drosophila) evolution revealed by mutation clocks. *Mol Biol Evol*. 21:36–44.

Thomson TM, Lozano JJ, Loukili N, et al. (14 co-authors). 2000. Fusion of the human gene for the polyubiquitination coeffector UEV1 with Kua, a newly identified gene. *Genome Res*. 10: 1743–1756.

Uchiyama I. 2006. Hierarchical clustering algorithm for comprehensive orthologous-domain classification in multiple genomes. *Nucleic Acids Res*. 34:647–658.

Vilella AJ, Severin J, Ureta-Vidal A, Heng L, Durbin R, Birney E. 2009. EnsemblCompara GeneTrees: complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res*. 19: 327–335.

Vogel C, Berzuini C, Bashton M, Gough J, Teichmann SA. 2004. Supradomains: evolutionary units larger than single protein domains. *J Mol Biol*. 336:809–823.

Wang W, Brunet FG, Nevo E, Long M. 2002. Origin of sphinx, a young chimeric RNA gene in Drosophila melanogaster. *Proc Natl Acad Sci U S A*. 99:4448–4453.

Wang W, Yu H, Long M. 2004. Duplication-degeneration as a mechanism of gene fission and the origin of new genes in Drosophila species. *Nat Genet*. 36:523–527.

Wang W, Zhang J, Alvarez C, Llopart A, Long M. 2000. The origin of the jingwei gene and the complex modular structure of its parental gene, yellow emperor, in drosophila melanogaster. *Mol Biol Evol*. 17:1294–1301.

Wang W, Zheng H, Fan C, et al. (14 co-authors). 2006. High rate of chimeric gene origination by retroposition in plant genomes. *Plant Cell* 18:1791–1802.

Wapinski I, Pfeffer A, Friedman N, Regev A. 2007. Natural history and evolutionary principles of gene duplication in fungi. *Nature* 449:54–61.

Weiner J, Beaussart F, Bornberg-Bauer E. 2006. Domain deletions and substitutions in the modular protein evolution. *FEBS J*. 273:2037–2047.

Weiner J, Bornberg-Bauer E. 2006. Evolution of circular permutations in multidomain proteins. *Mol Biol Evol*. 23: 734–743.

Wiedenhoeft J, Krause R, Eulenstein O. 2011. The plexus model for the inference of ancestral multi-domain proteins. *IEEE/ACM Trans Comput Biol Bioinform*. 8:890–901.

Xiao H, Friesen JD, Lis JT. 1995. Recruiting TATA-binding protein to a promoter: transcriptional activation without an upstream activator. *Mol Cell Biol*. 15:5757–5761.

Yanai I, Wolf Y, Koonin E. 2002. Evolution of gene fusions: horizontal transfer versus independent events. *Genome Biol*. 3:research0024.1–research0024.13.

Zhou Q, Zhang G, Zhang Y, Xu S, Zhao R, Zhan Z, Li X, Ding Y, Yang S, Wang W. 2008. On the origin of new genes in Drosophila. *Genome Res*. 18:1446–1455.

Zmasek CM, Eddy SR. 2001. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics* 17:821–828.

Zmasek CM, Eddy SR. 2002. RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics* 3:14.