Research Article:

**Evolution at the sub-gene level: gene architecture rearrangements in the Drosophila phylogeny**

Yi-Chieh Wu¹, Matthew D. Rasmussen¹, Manolis Kellis¹,²,*

1. Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA
2. Broad Institute of MIT and Harvard, Cambridge, Massachusetts, USA

*Corresponding author:
Manolis Kellis
32 Vassar St. D530
Cambridge, Massachusetts 02139
Phone: 617-253-2419
Fax: 617-452-5034
E-mail: manoli@mit.edu

**Keywords:** phylogenetics, gene fusion and fission, architecture evolution

**Running title:** XXX
Abstract

Although the possibility of domain shuffling 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 rearrangement occurs in as many as 25.8% of architecture families and involve a gene from 36.2% of FlyBase gene families, and thus, any comprehensive study of gene evolution in these species will need to account for and model gene rearrangements. Here, we present a new computational model and algorithm for reconstructing gene evolution at the domain level. We develop a method for detecting homologous regions between genes and a phylogenetic algorithm for reconstructing maximum parsimony architecture scenarios, where our architecture model allows us to trace the generation, duplication, loss, merge (fusion), and split (fission) of these regions. Using this method, we find that genes involved in fusion and fission are enriched in signaling and development, suggesting that gene rearrangements and region reuse may be crucial in these processes. We also find that fusion is preferable to fission and that duplication is essential to fusion and fission events, with 92.5% of fusions and 34.3% of fissions retaining a pre-fusion/fission architecture, thereby allowing ancestral genes and functions to be retained. We further consider possible mechanisms for gene rearrangements and provide a catalog of ~9000 genes (across nine sequenced species) that result from these mechanisms. These results provide a better picture of evolution at the sub-gene level and offer several insights into how new genes and functions arise in evolution.

Author Summary

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 [1, 2, 3], while gene loss plays a similarly important role in shaping genomic content [4, 5]. These events, as well as several others such as horizontal transfers, gene conversion, and gene fusion, interact together to generate gene families, clusters of orthologous and paralogous genes that share detectable common ancestry. By studying the genetic sequences of a gene family, one can infer many of the evolutionary events 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 (Figure 1B). In the simplest case, these two trees 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 has occurred. One can infer these events by combining several computational methods. Phylogenetic methods, such maximum likelihood [6] or neighbor joining [7], can be used to reconstruct a gene tree and species tree from molecular sequences, and special algorithms called reconciliation methods [8, 9] 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 carried across both sizable clades and whole genomes, in a research field called phylogenomics [10, 11]. Many computational methods have been developed for detecting and reconstructing gene families as well as their events [2, 12, 13, 14, 15, 16, 17, 18, 19]. This has led to a better understanding of how evolution shapes the gene content of many different species such as yeasts [15, 16], flies [4], and vertebrates [17]. (XXX: may need microbial citations here)

Despite the sophisticated underlying models in these methods, a common assumption is to consider a gene as evolving as a single unit. However, it is just as possible that duplications, losses, and other events can occur at the sub-gene level. Additionally, events such gene fusion and fission also complicate the 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 an equally important role in generating novel genes and functions; however, in order characterize these events systematically, new models and methods are needed.

There is evidence that these more complex fusion and fission events occur quite frequently. For example, jingwei is a chimeric gene found in Drosophila yakuba that arose through the fusion of the two genes yande and Adh (alcohol dehydrogenase). Although a fusion of genes is likely deleterious, several factors in this case have contributed to jingwei’s retention. First, the parental genes of this fusion were kept intact, since yande is itself a recent duplication of yellow-emperor and the Adh portion of jingwei is a retrotransposed copy of the original Adh [20, 21, 22]. Thus, the parental genes (yellow-emperor and Adh) can retain their functions (nuclear mRNA splicing and alcohol binding), while the fused gene is free to acquire a novel function (more specific binding for long chain alcohols) [23]. Second, jingwei’s retention is likely aided by the fact that it has inherited the promoter sequence of yande, as the promoter-lacking Adh sequence retrotransposed into the 3’ end of jingwei. Other examples of exon shuffling in Drosophila include the fusion genes Adh-Twain [24], Adh-Finnegan [25], siren [23], sphinx [26], and Quetzalcoatl [27]. More broadly, it has been indirectly estimated (through intron phase correlations) that as
many as ∼19% of exons in eukaryotic genes might have been formed by exon shuffling [28]. However, large-scale methods for the systematic identification of these genes and the reconstruction of their ancestral events are still lacking. In particular, in order to fully characterize the impact of fusion and fission events across the genome, a phylogenomic approach is needed that operates at the sub-gene level.

Studies of genes at the sub-gene level have often focused on domains, which are evolutionary, functionally, and structurally conserved units of proteins. Bornberg-Bauer et al. [29] provides a thorough review on existing knowledge and approaches for studying the evolution of domains and their rearrangements, though only recently have studies been conducted at the genome level. Of these, many studies focus on quantitative analysis of domain arrangements [30, 31] in order to identify trends such as fusions are preferable to fission [32], the number of neighbors per domain follows a power law [29] (though this could be attributed to limited coverage [33]), and domain combinations are conserved in that there are fewer combinations than would be expected from random domain shuffling [29]. Multi-domain proteins have also been studied in the context of sequence similarity networks for determining gene families [34, 35, 36]. Initial studies to detect gene fusions looked at widely divergent species and performed pairwise protein comparisons to find non-transitive BLAST hits [37, 38, 39]; a few recent studies have focused on identifying recent fusion and fission events within clades such as bacteria [40], fungi [41], and Drosophila [42, 43], or on studying the formation of specific chimeric genes [44, 45, 46].

In addition to these quantitative studies, there has been a recent growth in phylogenomic studies of domain evolution, with a number of architecture-aware phylogenetic methods developed in the last decade. Perhaps the first such method was developed by Snel et al. [32], who detected sub-gene regions based on BLAST hits and generated phylogenetic trees for proteins but only considered ancestral architectures consisting of one fused protein or two split proteins. With the advent of domain databases such as Pfam [47], SCOP [48], SMART [49], and CDD [50], all later phylogenomic studies shifted to rely on underlying domain models rather than de novo identification of domains through sequence similarity. Like the earlier studies, however, they also looked at rearrangements across widely divergent species, e.g. across all three kingdoms (XXX: domains?) of life. These methods can be broadly classified into three categories.

Rather than looking at gene evolution, the approaches in the first group simplify the problem to look only at the presence or absence of architectures within each genome, thus ignoring both architecture count and sequence information. For example, to study convergent evolution of domain architectures, Gough et al. [51] assumed that an architecture present in one genome and not another must have been lost or gained. In later works, Kummerfeld and Teichmann [52] and Fong et al. [53] modeled architecture rearrangements and developed dynamic program-
ming (DP) maximum parsimony (MP) algorithms for inferring ancestral architectures by mapping events between nodes of a known species tree such that some cost metric, e.g. the total number of fusion and fission events, was minimized.

Another group of methods looks explicitly at gene evolution by accounting for the count of each architecture type, but the added complexity was often compensated by simplifications elsewhere in the model. Furthermore, these approaches have generally leaned towards theoretic formulations and only been applied to a limited amount of biological data. For example, while they did not explicitly reconstruct phylogenetic trees, Przytycka et al. [54] described a graph-theoretic approach for analyzing the evolution of multidomain proteins. Whereas all previous methods ignored sequence after domains had been assigned to genes, Behzadi and Vingron [55] utilized this information by using a three-step approach: existing phylogenetic algorithms were used to reconstruct domain trees, these were reconciled to the species tree, and sets of domains at each species node were partitioned such that the total number of fusions and fissions was minimized. However, despite considering the combinatorially simpler problem of domain combination rather than architecture, they considered the generation of each child architecture separately, used a simple counting statistic to trace the number of fusions and fissions, and relied on computationally expensive simulated annealing techniques. More recently, Wiedenhoeft et al. [56] developed a plexus data structure for modeling the evolution of domain combinations, including generation, duplication (including domain repeats within a gene), loss, and fusion, though they do not consider the problem of reconciliation and thus may not properly count events across the species tree.

The final group of reconstruction methods takes an alternative view of architecture evolution by mapping domain level events onto existing trees. They are also the only known methods to consider a bootstrapping strategy. Forslund et al. [57] generated a phylogenetic tree for each domain, then used a DP MP algorithm to map domain insertions and deletions onto these individual trees, looking for agreement between architecture creation events only after all mappings. A similar approach was developed (XXX: Durand), though they used a reference gene and (XXX: ML?)-reconciliation to map domain insertions and deletions.

In this work, we continue along these recent methods and extend phylogenomics from genes to sub-gene domains, or more generally to regions, where a region (XXX: is this the name we want? maybe atom?) is simply a gene subsequence. By focusing on regions, 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 regions and the duplication, loss, and rearrangement of existing regions.
We present three major contributions to sub-gene phylogenomics:

- We present a method for identifying homologous regions for a family of closely related species. Our approach uses sequence similarity to determine regions, and therefore is not limited to existing domain databases which may be biased towards, for example, structurally determined domains. We show that the resulting homologous regions reflect underlying mechanisms for region rearrangement; in particular, they are produced through exon shuffling and tend to keep functional domains as a single unit.

- We develop a model for architecture rearrangements (Figure 1E) and an accompanying maximum parsimony algorithm STAR-MP for determining architecture evolution (Supplemental Figure S6). These are less restrictive than previous phylogenetic approaches and can be considered a cross between them, retaining the advantages of each. In particular, we assume a known species tree, as the added information can improve gene tree reconstruction; we trace gene evolution rather than architecture evolution, allowing us to explicitly model duplications; we do not rely on a reference gene or domain but instead view regions as the basic building blocks of genes rather than secondary structures that enter or exit other gene or domain families, allowing us to trace the evolutionary history of genes related through any subsequence within a single reconstruction; we incorporate sequence information during phylogenetic reconstruction; and we consider the statistical support of our reconstructions through bootstrapping. (XXX: The STAR-MP software is written in Python and is available for download at http://compbio.mit.edu/star-mp/.)

- We identify several biologically meaningful trends in fusion and fission events. In particular, we show that these events reflect adaptability and allow for functional complementarity. In the former, architecture rearrangements allow a species to reuse regions across multiple genes rather than independently generating them for each gene, which may be particularly important when a region is associated with a function crucial to several pathways as in signaling. In the latter, split gene forms (e.g. the parents of merge events or the children of split events) often have related functions, where the split forms have merged into a single gene, perhaps for tighter co-regulation, in the case of fusion, or a single function has separated into two related genes in the case of fission. (XXX: re-emphasize biological results mentioned in the abstract? mention mechanisms, etc?)

To demonstrate the sensitivity and robustness of our methods, we consider eukaryotic species that are evolutionarily 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, identi-
fying only merge and split events that have occurred in recent history; we focus our analysis on the *Drosophila* clade (Figure 2), as it has a dense phylogeny, a relatively recent (~60 million year old) history [58], and includes both close and distant species. Furthermore, at least 47 putative chimeric genes have been identified within *D. melanogaster* [42, 43], and it has been estimated that ~30% of the new genes in the *D. melanogaster* species subgroup are chimeric [42].

**Methods**

We present a novel phylogenomic pipeline for the architecture-aware reconstruction of gene evolution (Figure 3). This has three main stages: (1) identifying regions and region 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.

(XXX: need to maintain the difference between model and method - possibly move graph representation of gene architecture from supplement to main text. The graph representation is already briefly mentioned in the overview figure.)

**Definitions**

Due to fusion and fission, a gene may have segments along its length that differ in homology and phylogeny. Therefore, we must introduce several new terms to describe the possible relationships between such genes. Our primary unit of evolution is the *region*, which is a gene subsequence such that each gene may contain one or more non-overlapping regions. These regions may share homology with other regions present in other genes. We call a cluster of homologous regions a *region family*, defined as the set of regions that descend from a single region in the last common ancestor (LCA) of all species under consideration. Each gene is said to have an *architecture*, which is an ordered list of region families. We also generalize the concept of a gene family to that of an *architecture family* which contains the maximal set of genes connected by region homology. Finally, a multiset of architectures can be assigned to each species, and we can construct *architecture trees*, which are extensions of gene trees and more accurately described as directed acyclic graphs that show the evolutionary relationship between genes within an architecture family. In this work, we simplify this concept to an *architecture scenario*, which is simply a species tree with architectures (type and count) mapped to each node and evolutionary events mapped to each branch. In particular, in reconstructing architecture scenarios, we will assume a known species tree and infer ancestral
architectures and events; no reconciliation is required. All trees within this work are rooted phylogenetic trees in which the leaf nodes represent the regions or architectures in extant species, and the internal nodes represent ancestral regions or architectures in ancestral species.

We allow the following evolutionary events for transforming architectures: generation, in which a new region is created; duplication, in which an existing region is duplicated; loss, in which an existing region is lost; merge, in which two regions that appeared at the ends of two separate architectures are joined as neighbors in a single gene; and split, in which two regions 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 region can be generated at most once. This is similar to the assumption used in Dollo parsimony in which a single generation in the last common ancestor followed by (multiple) losses is more likely than multiple generation events. Note that these definitions require that generation, duplication, and loss occur at the region level. For example, generating a sequence of multiple regions is only possible through generation of the component regions. Our assumption treats regions as the basic building blocks of a gene and implies that the generation, duplication, and loss of larger sequences (as measured by the number of regions) incurs a higher cost.

(XXX: work more on) As we will see later, this choice of multi-region generations, duplications, and losses as multiple events allows for a simpler reconstruction methodology. However, it may both contradict and support underlying biological processes; for example, the loss of multiple regions can occur with the introduction of a single stop codon, but the loss of multiple regions may also be less probable since it likely renders the parent gene dysfunctional. Similarly, for simplicity, we treat a merge or split as a single event that incurs a cost independent of duplications and losses when the underlying mechanism may actually involve duplications and/or losses (see Systematic detection of gene fusion and fission by mechanism).

Genomic sequences and species phylogeny

Analysis was performed on nine species within the Drosophila genus: D. melanogaster, D. yakuba, D. erecta, D. ananassae, 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 [59] (Figure 2).
Identifying regions and region families

We ran pairwise all-vs-all BLASTp [60] 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 [61], and the best hit between each query and subject pair was retained. These were re-filtered by $e$-value and percent identity, and short alignments ($< 50$ aa) and promiscuous hits (genes with $> 80$ hits) were removed. A list of potential region boundaries was then found using the residue correlation matrix as in the ADDA algorithm [62] (resolution = 10 aa, minimum region length = 30 aa), and boundaries within 30 aa of a LALIGN alignment boundary were retained. The resulting region instances were clustered into region families through OrthoMCL with default parameters [34], where the nodes represent region instances and edges are weighted by the bitscore of the LALIGN hit multiplied by the relative overlap of the regions. (XXX: Note that these steps can be replaced by a database search that matches gene sequences against a known database of domains and simultaneously detects the domain boundaries and domain families.)

In this work, we are mainly interested in studying merge and split events that occur across multiple species; thus, we filtered our region families to focus on these events. Therefore, any region families that appear in only a single species were discarded. Also, to allow for a more efficient algorithm, we collapsed tandem duplicated regions to a single copy, and required that a region family can appear at most once within an architecture.

Clustering into architecture families

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

From the region adjacency graph, we discovered several highly promiscuous region families that occur in diverse sets of genes. These region 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 region family. Therefore, we choose to analyze promiscuous region families in a separate analysis (Supplemental Methods S1) and excluded them from our reconstructions. Specifically, region families were removed prior to clustering if they had more than 6 neighbors; this removed $< 0.21\%$ of all regions and $< 0.38\%$ of the regions 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 connected regions and another species has a gene with at least one of these regions unconnected.

**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 region counts. Rather than doing these tasks simultaneously, we recognize that because generation, duplication, and loss events occur at the region level, we can reconstruct the generation, duplication, and loss history of each region independently of all other regions, use these region phylogenies to determine ancestral region counts, and finally incorporate merge and split events when inferring region groupings into architectures. Thus, we adopted a three-stage approach to architecture scenario reconstruction.

In the first stage, we reconstructed the phylogenies of each region family to produce *region trees*. This was done by taking the peptide sequences of each region family, aligning them with the MUSCLE software package [63], then reverse translating the result into a (codon-aligned) nucleotide alignment. Region trees were then reconstructed from each nucleotide alignment using the SPIMAP program [19] configured with model parameters previously determined for the *Drosophila* clade [19], 100 pre-screen iterations, and 50 iterations.

In the second stage, regions trees were split into subtrees containing only descendants of a single common ancestor within or after the root of the species tree (i.e. proper region families). This was achieved by reconciling each region tree to the species tree using maximum parsimonious reconciliation (MPR) [64, 65] and then removing any duplication nodes predating the species tree root (pre-root duplications). Each resulting subtree was then rerooted and reconciled repeatedly using MPR until all pre-root duplications were removed.

In the third stage, we reconstructed architecture scenarios for each architecture family by combining all of its region 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 region trees to infer the ancestral region copy numbers. What remains to be reconstructed is how the ancestral regions combine to form ancestral architectures and what events are responsible for their evolution.

We achieve this reconstruction using a novel maximum parsimony method called STAR-MP (Species Tree informed Architecture Reconstruction - Maximum Parsimony, Supplemental Figure S6), 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.

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 perform a post-order traversal of the species tree, generating a set of possible architectures for each node by finding all partitions of the available regions, 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 is then used to find the minimum cost-to-go (e.g. minimum total cost along all descendant branches) of assigning the parent architecture. This is repeated until the root of the species tree is reached, at which point the minimum-cost architecture is assigned to the root. In the backward phase, we backtrack down the tree to determine the most parsimonious architectures and events at all the internal nodes and edges, respectively. Ties were broken arbitrarily; thus, the maximum parsimonious architecture may not be unique. For more details, refer to Supplemental Methods S11.

To measure uncertainty in our reconstructions, we bootstrapped our architecture scenario reconstruction. Each region family had 100 region trees reconstructed using SPI Mam on 100 resampled nucleotide alignments. From this set, regions trees were sampled with replacement to be reconciled and analyzed by STAR-MP 100 times, thus generating 100 bootstrapped architecture scenarios.

**Results**

(XXX: Think about reordering subsections)

Using our pipeline, we found 22,861 region families combining in 14,418 architecture families. 70.4% (10,144) of these architecture families contain only a single region, and (XXX: say something about how this number is “surprisingly” high?) 28.5% (4107) contain a merge or split. These “merge/split” families consist of 12,324 region families covering 45,282 genes, and involve at least one gene from 36.2% (4457/12,324) of FlyBase gene families. (XXX: In general we need to be careful about how your filtering rules affect these numbers. We should always report the percentage that is easiest for the reader to understand. – All numbers here use only the filters listed in the methods, e.g. no filtering for neighboring regions, alternative splicing, etc.)

(XXX: Moved from trends section - some of this belongs in methods?) Architecture scenarios were reconstructed for 3882 families (with 10,448 region families covering 39,476 genes), of which 1064 (27.4%) had ties;
the remaining 5.5% of families had many region 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 (XXX: runtimes are better described in methods). Analysis of architecture scenarios (see Trends in evolutionary events and Effect of genome annotation errors on reconstruction) considers non-bootstrapped reconstructions; bootstrapped architecture scenarios were well-supported, with the majority (63.2%) of scenarios having a single reconstruction and event counts having a low standard deviation relative to its mean (<0.035), thus demonstrating the robustness of our reconstruction methodology. (XXX: Brief bootstrap outcome might be better described in methods)

Validation

Input validation

A significant challenge of reconstructing architecture evolution is dealing with errors in extant genomes, e.g. 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 regions 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 affect 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 36.2% (1486) 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% (2947) of genes in merge/split families are at the ends of scaffolds compared to 4.85% (6592) overall, meaning that we possibly find inflated counts for the number of merges and splits. In the latter case, we would expect near 100% identity in the sequences. Analysis of the sequences using gene spans with 2000 base pairs added upstream and downstream reveals 7.31% (300) of the merge/split families have possibly undercollapsed scaffolds (scaffolds contain undercollapsed genes with ≥ 98% identity, Supplemental Methods S2). 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, Supplemental Methods S3). We found that only 0.92% (0.52%) of EST (mRNA-seq) supported neighboring gene models also had an EST (mRNA-seq) spanning both neighbors, suggesting a low rate of introns misannotated as intergenic regions. The lowest intron annotation error rate was in the well annotated \textit{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 (Supplemental Tables S1), 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 regions, e.g. spurious gene breaks, not intergenic regions misannotated as introns, e.g. missed gene breaks.

\textbf{Methods validation}

Most methods within our phylogenomic pipeline (e.g. residue correlation matrix, OrthoMCL, SPIMAP) have been evaluated in their respective works [62, 34, 19]. To evaluate the last step in this pipeline, our architecture scenario reconstruction algorithm STAR-MP, we simulated region evolution, where simulation parameters were inferred using the MP architecture scenarios reconstructed from real data. Note that this reliance on MP reconstructions means that our simulations underestimate 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 region families, the number of regions per region family, and the number of connected regions 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 regions (generation/duplication/loss), edges (split), or architectures (merge), and was discarded if it was impossible with the given starting architecture.

We ran simulations of generation/duplication/loss/merge/split rates at 1X, 2X, and 4X the estimated true rates. The generation rate was kept constant at 1X, the other rates were combined into seven different settings (Figure 4), and for each rate setting, 1000 architecture scenarios were simulated. As in actual pipeline, the ancestral counts for each region and the architectures at the extant species were provided as input to STAR-MP. In general, STAR-MP has higher precision than sensitivity for any given event, and performance tends to degrade as the event rates increase and the true architecture scenarios become more complicated. Indeed, part of the decrease in sensitivity can be attributed to trying to explain more complex architecture scenarios with a conservative MP algorithm. STAR-
MP also tends to have higher generation, duplication, and loss performance than merge and split performance, and as expected, generation, duplication, and loss performance is consistent across various merge and split rates. The events generation, duplication, and loss had 100% precision since they are inferred from ancestral region counts, which are provided directly as input. Interestingly, merge performance is typically higher than split performance. Further investigation shows that low split performance can be attributed to cases in which an architecture is partially lost. Here, the true reconstruction is a merged parent architecture undergoing a split and loss to result in the surviving sub-architecture; however, STAR-MP tends to reconstruct a split parent architecture so that only a loss is needed to produce the surviving sub-architecture.

**Region boundaries are driven by selection: comparison with domains and exons**

Here, we look to two external lines of evidence, functional domains and exons, to determine how regions are formed. Note that our method for finding regions depends only on sequence similarity and is not defined *a priori* by domains or exons as with other gene rearrangement methods.

We find that 75.6% of all regions are equal to or larger than Pfam [47] domain definitions (Figures 5A and 5B), indicating that our region detection algorithm tends to avoid over-fragmentation. Furthermore, most regions and domains are of similar size, as evidenced by the distinct peak near 100%, and similarly, most region boundaries are close to domain boundaries (Figure 5C).

Comparison between regions and exons reveals similar trends (Supplemental Methods S4, Supplemental Figure S1), with many cases of single region-single exon or single region-multiple exons, and a large percentage (33-42%) of regions lying precisely at an exon boundary (peak at zero distance in Supplemental Figure S1a). To study this effect further, we looked at the number of exon-bordering regions (Supplemental Table S2) and at intron-phase correlations (Supplemental Table S3). We defined an exon-bordering region as a region in which both boundaries are within ±10 residues of an exon boundary. The unusually high number of exon-bordering regions (observed = 100,974; expected = 2138; fold = 47.23; \( p < 2.23 \times 10^{-308}, \chi^2 \) test) indicates exon shuffling as a prominent mechanism of region rearrangement.

Exon shuffling was 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 region is labeled with the phases of its flanking introns. The splice-frame rule [66] states that the phases of introns flanking regions tend to match, as this prevents frameshift mutations after exon shuffling events. Similar to previous analyses [67, 68, 69], we find that symmetrical intron
phases are enriched (O = 83,394; E = 35,003; fold = 2.38; $p < 2.23 \times 10^{-308}$, $\chi^2$ test) and non-symmetrical intron phases are depleted (O = 17,580; E = 65,971; fold = 0.27; $p < 2.23 \times 10^{-308}$, $\chi^2$ test). Furthermore, most of the enrichment in symmetrical intron phases is due to the presence of 0-0 regions; 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 (Supplemental Table S4), fold enrichments and depletions are dramatically increased for regions (e.g. fold values: exon-bordering domains = 2.32, symmetrical intron phases = 1.79, non-symmetrical phases = 0.58, $p < 2.23 \times 10^{-308}$, $\chi^2$ test), and we find an abundance of 0-0 regions and a lack of 1-1 regions compared to previous analyses. These discrepancies are expected, as previous works used domain definitions produced across many genomes, whereas our regions are detected using data only across the nine *Drosophila* genomes. Regardless of whether domains or regions are used, these results suggest that regions (and domains) are produced through the shuffling of exons; here, a mutational mechanism is made apparent through region (domain) detection. (XXX: try to shorten this paragraph)

An alternative explanation for the correlation between region and domain boundaries correlation might be their common correlation with exon boundaries. Thus, we tested whether region boundaries are depleted within domains, which would suggest that regions tend to maintain domains as a unit more so than would be expected by exon distributions. We find that 7.1% (29,096/410,463) of introns are within ±10 residues of any region boundary, whereas within domains, this percentage decreases to 3.0% (4451/146,205), supporting our expectation that region boundaries respect domain boundaries (fold = 2.33, $p < 2.23 \times 10^{-308}$, hypergeometric test). (XXX: It is tough to understand how this related to the question posed)

(XXX: This caveat should be explained in its own paragraph, should we also cite the tblastx work?) We must be careful in our analysis, however, as we found region boundaries using protein sequences. This implicitly assigns any region boundary that occurs within an intron to the nearest exon boundary; thus, a large number of long introns would result in many region boundaries falling exactly on exon boundaries. (XXX: All previous studies use protein sequences in analysis, though that just means we all have the same flaw.)

**GO term enrichment reflects adaptability**

In this section, we address whether certain functions are more likely to be involved in merge and split events. After correcting for possible biases (Supplemental Methods S6), we find 8 GO terms to be enriched across families with merge/split events compared to families without merge/split events ($p < 0.001$, hypergeometric test, FDR 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 require 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 crucial in signaling and development [70, 71], explaining the enriched GO terms in these categories.

(XXX: reword since copied from Pfam/refs?) For example, consider the TBP (TATA-binding protein) 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 region. A study of TBP genes hypothesizes that the N- and C-terminal regions may have evolved independently of each other and fused together [72]. 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 [73]. Metazoans may have evolved multiple TBPs to accommodate the vast increase in genes and expression during development and cellular differentiation [74].

**Fusion and fission allow for functional complementarity: a look at protein-protein interactions**

It has been shown that regions that merge or split tend to occur in genes with related functions [37, 38, 75]. 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. One reason for this trend is that the fusion of neighboring genes allows for tighter co-regulation [29]. 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 1222 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 regions: the first set contains the regions in gene 1 but not in gene 2, and the second set contains the regions in gene 2 but not in gene 1. To be called a *gene partner*, at least one pair of regions, 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 find that 138 gene partners have both genes annotated with GO terms, and of these, 114 (82.6%) share at least one GO term. If we select random gene partners (to control for length bias, these partners are selected from the set of 208 genes that form the 1222 partners), we observe that 61.8% share a GO term on average. This suggests that genes are more likely to have related functions if they have regions that merge or split (fold = 1.34, $p < 0.001$, (XXX: random sampling test?)), though the cause-and-effect may be the reverse. (XXX: Reference Supplemental Methods S7 for results using PPI?)

**Trends in evolutionary events**

Our architecture scenarios that involve region merges and splits cover 4107 architecture families, 12,324 region families, and 45,282 genes. However, many of these families have very simple scenarios. Most (2295, 55.9%) contain only two regions (Figure 6), and many (1007, 24.5%) contain one gene in each of the nine species. These single gene families frequently consist of distinct subtrees, one with a single region A and another with merged architecture AB, implying a single generation and merge of region 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 revisit in the next section.

Using our reconstructed architecture scenarios, we studied the distributions of each of our events (Table 3, Figure 7, Supplemental Figure S4).

For generation events, we find that most regions (8339/10,448; or 79.8%) are generated at the species tree root (Table 3) and are 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 non-coding sequences accounts for 11.9% of new genes [42], suggesting that partial gene origination may not be rare [3].

For duplication and loss events, we observe 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 [19] (XXX: other citations?). This arises due to paraphyletic regions (regions 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 regions trees that are inconsistent with the species tree so that during reconciliation, a single duplication is compensated for with multiple losses.

Lastly, for merge and split events, a comparison of their counts reveals a 0.86:1 merge-to-split ratio, which at first seemed inconsistent with previous studies suggesting that fusion is preferable to fission by a factor of 2.6 –
5.6 [32, 52, 53]. However, one key difference in this analysis is that we measure individual events, as opposed to simply observing the presence of fused and fragmented extant genes, and we are measuring events over a smaller, higher resolution time scale (the 62 myr Drosophila clade vs. all three kingdoms (XXX: domains?) 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 architectures AB and A) and partial gene loss (e.g. architecture AB to architecture A). We consider 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 shows that many splits are due to such partial duplications and losses; if we consider only “simple” merges and splits that are unaccompanied by generation, duplication, or loss events, the merge-to-split ratio becomes 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 find is that merge and split events tend to co-occur within families and regions. There are 1264 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, 2419 region families are involved in both merge and split events (42.9% of the 5645 region families that undergo a merge, 34.3% of the 7049 region families that undergo a split). This suggests that regions that undergo a merge or split event are more likely to undergo further rearrangement (compared to the 22,861 region families in Drosophila, fold = 1.39, hypergeometric 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 (non-generated) regions, and most (92.5%) retain at least one pre-merge 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 find that most (52.6%) split events occur with the loss of a resulting split region, and few (34.3%) retain the pre-split architecture. (XXX: Removed mentions that merges between existing regions are more likely to retain pre-merge architectures (94.9% vs. 85.8%, hypergeometric test, \( p = 3.12 \times 10^{-23} \)), and splits without accompanying losses are more likely to retain pre-split architecture (68.7% vs. 3.3%, hypergeometric test, \( p < 2.23 \times 10^{-308} \)). These are indicated in the table - necessary to put in text?)
Effect of genome annotation errors on reconstruction

We find that 57.4% of all merge events and 78.9% of all split events occur along a branch leading to an extant species. This could suggest that merge and split events tend to be lineage-specific, as found in previous studies of *Drosophila* [42, 43], or it could be an artifact of our pipeline arising from poor gene models and architecture annotations. For example, 4.7% of all merge events occur along the branch leading to *D. melanogaster* despite the branch accounting for only 2.9% of the total branch length within the species tree. However, *D. melanogaster* also has the lowest gene model error rate (Table 1) and likely the fewest number of spurious gene breaks. Thus, this large count 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 where 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 due to poor genome annotation. Though we have previously validated our sequence input, we also decided to consider a highly conservative set of architecture families, which we define 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. Filtering these families resulted in a set of 2506 families (61.0% of original set) with 6120 regions (49.7%) covering 21,758 genes (48.0%). Of these, 2492 families with 6022 regions covering 21,518 genes had reconstructed architecture scenarios. Note the two-fold decrease in the number of sequences represented. This is expected as our conservative set likely removes many true examples of gene fusion and fission; for example, all trees with neighboring genes merging or a gene splitting into two neighboring genes are removed though both of these are valid potential mechanisms. Though this conservative set changes ratios and folds, all results within the previous section hold (Table 3, Supplemental Figure S5).

Detection of previously identified chimeric genes

Zhou et al. [42] and Rogers et al. [43] previously identified 47 unique chimeric genes in *D. melanogaster*. 21 of these were also identified by our algorithm (Supplemental Table S6), yielding a sensitivity of 44.7%. However, Zhou et al. [42] 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 may not be detected by our pipeline since a gene region must have a hit for it to propagate through our region detection algorithm, and our use of protein sequences eliminates any
possible hits to intronic or integenic sequences. If we consider 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 (1), no hits found satisfying the percent identity threshold (1), frameshift mutations (1), overlapping alignments (2), or under-clustering of regions into region families (1). We will not expand on the first two reasons since they are a consequence of the BLAST step in our pipeline, and we chose thresholds consistent with previous studies in phylogenomics [14]. Similarly, regarding the last reason, we chose a clustering threshold for OrthoMCL consistent with previous studies [34].

Both Zhou et al. [42] and Rogers et al. [43] used BLASTn to compare CDS sequences, and they used different filters, e.g. 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 eliminates our ability to detect frameshift mutations. Investigation of nucleotide alignments (Supplemental Methods S8) 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%. 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. (XXX: The phylogeny should allows us to really tell whats going on) Aside from sequence changes in the datasets that could have caused non-overlapping alignments to now appear as overlapping, remember that we also extended our alignments using LALIGN, whereas Zhou et al. [42] and Rogers et al. [43] used BLASTn alignments. Manual inspection of the alignments suggests that the full overlapping alignments are correct, and the two cases correspond to non-chimeric genes. (XXX: Nice! which paper are these overlapping genes from?)

**Systematic detection of gene fusion and fission by mechanism**

In this section, we consider possible mechanisms for generating new gene architectures that require merges and/or splits (Figure 8), show a concrete example of the mechanism, and determine how often each mechanism occurs within *Drosophila* (Supplemental Methods S9, catalog of genes by mechanism is available online). (XXX: Necessary?) For each example, we provide (A) the MP reconstructed architecture scenario, with the merge/split event of interest shown in an inset, (B), a genome level view of the genes involved in this merge/split, and (C), a hypothesized mechanism for the merge/split. The architecture scenario shows the architectures (region arrangements
and count) in each species and the events along each branch, and the genome level view reveals any synteny and
shows each gene in three ways: the top view provides the regions within the gene, the middle view provides the
gene directionality, and the bottom view provides the gene model.

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 find that
1681 regions and 6713 genes (16.4% and 17.2% of the regions/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 find 274 (236) of the above genes are inconsistent
with ESTs (mRNA-seqs), 5863 (4534) genes have no ESTs (mRNA-seqs), and 576 (1943) 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
Figure 9) may be the result of large-loop mismatch repair or replication slippage [43]. We find that 32 regions and
19 genes (0.3% and 0.05%) possibly undergo/result from these mechanisms.

The second mechanism was introduced with the case of jingwei (Figure 10), 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. 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 find that 1904 regions and 2023 genes (18.5%
and 5.18% of regions/genes participating in merge/split events) potentially undergo/result from this mechanism.
In comparison, previous studies have found that retrotransposition accounts for 12.2% of chimeric genes in D.
melanogaster [42].

The third mechanism involves segmental duplication followed by differential loss and was observed in the
monkey king family [76]. Though we could not find this example in our dataset as the events occur in a sister
group of D. melanogaster not included in our nine species, we find that 60 regions and 79 genes (0.6% and 0.2%
of the regions/genes participating in merge/split events) undergo/result from this mechanism. An example is the
evolution of the rhea family (Figure 11).

**Fusions and fissions are supported by transcript evidence**

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 (Supplemental Section S10).
We find 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 (Supplemental Table S7). While 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. (XXX: what else?)

**Discussion**

We have presented a novel maximum parsimony algorithm STAR-MP for tracing architecture evolution, and shown that our pipeline is able to capture the generation, duplication, loss, merge, and split of regions to form new genes. Unlike conventional gene tree reconstruction methods, our approach incorporates gene architectures and is thus able to model how genes across gene families may be related, as indicated by the presence of similar regions or architectures. Also, unlike most architecture-aware phylogenomic analyses, our approach finds gene regions de novo rather than relying on external domain models, and our reconstruction pipeline traces gene evolution while incorporating sequence information and providing statistical bootstrapping support.

Our results corroborate previous findings that merges are preferable to splits, and we have shown that merge and split events tend to occur with duplications so that original architectures are retained. An associated question is whether merges and splits are enriched in genes that are alternatively spliced, and how often an alternatively spliced form carries the original architecture. We did not perform such analysis in our results since regions were only mapped in the longest polypeptide per gene, and only *D. melanogaster* is well-annotated with alternatively spliced forms.

A major problem in accurate gene tree reconstruction is the lack of information in sequence data; this is exacerbated when constructing region trees due to the use of even shorter gene subsequences. We have attempted to mitigate such errors by using the species-aware program SPIMAP.

(XXX: Reword the future as possibilities and as next steps for the field not necessarily for ourselves) Like many approaches in sub-gene phylogenetics, we do not propagate sequence data through the entire pipeline, and we rely on parsimony rather than a more principled probabilistic methodology. These weaknesses suggest that the field could move to distance-based or probabilistic reconstruction methods, which we believe will better capture the evolutionary history of architecture families. (XXX: liken to gene trees moving from simple MP to more complicated models?) In particular, this would allow for the modeling of branch lengths within an architecture tree (rather than an architecture scenario) and thus place evolutionary events at specific timepoints within the
species history. This would also allow the simultaneously modeling of both sequence and architecture evolution, rather than the current approaches of utilizing sequence to reconstruct region trees then either using architecture to reconstruct architecture scenarios or using reconciliation to determine region insertions and deletions.

A major bottleneck of architecture reconstruction algorithms is the enumeration of possible architectures, which can use both the order of regions 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 relies on heuristics to limit the set of parent architectures, a using a maximum parsimony approach, it is 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 regions 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 region inversion are required for accurate architecture reconstruction.

We have applied our analysis to the *Drosophila* clade; analysis of other well-sequenced clades such as *Candida* may provide further insight into gene architecture rearrangements.

**Acknowledgements**

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 NSF Graduate Research Fellowship to Y.W. and NSF CAREER award 0644282 to M.K.

**References**


Figure 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 sub-gene level. (B) In conventional phylogenetics, genes that descended 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 (∗), loss (×), and speciation (colored blocks representing domains) events are inferred through the reconciliation 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 sub-gene 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 regions. This allows the reconstruction of gene histories to be architecture-aware, with an architecture scenario depicting more complicated events such as merges (▽) 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 reconciliation is required, and speciation events are not modeled. In this example, the joint histories of the red and teal regions are determined, including their recent merge in the branch leading to species A, corresponding to the formation of chimeric gene a2. (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 regions and edges representing neighboring regions (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.
Figure 2. Species tree, abbreviations, internal node numberings, and phylogeny of the *Drosophila* clade. The phylogeny of 9 *Drosophila* species used in our analysis, as estimated by Tamura et al. [59].
Figure 3. **Overview of our phylogenomic pipeline.** 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. In the first stage, gene sequences are compared across species, region boundaries are found, and regions are clustered according to similarity, resulting in a set of homologous region families. In the second stage, a region adjacency graph is constructed based on these region families, with an edge between any two region families if at least one region instance from each family are neighbors in the same gene. Connected components of this graph define the region families to be clustered into a single architecture family. In the third stage, architecture scenarios are reconstructed for each architecture family based on a three-step procedure in which the region trees are reconstructed based on multiple sequence alignments of each region family, these region trees are reconciled to determine ancestral region counts, and the region counts, extant architectures, and known species tree are used to reconstruct the ancestral architectures and ancestral events along each branch.
Figure 4. Sensitivity and precision of STAR-MP in simulation under various event rates. Error bars show performance loss due to ties in the MP reconstruction, e.g. 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.

Figure 5. Correlation of regions and domains. (A,B) For each region, the overlap for regions incompletely covered by domains, and the relative size (region length/domain length) for regions completely covered by domains, was found. The distribution and cumulative distribution are shown. Bin size = 10%. (XXX: may only need (a) and not (b)) (C) For each region boundary, the distance to the closest domain boundary was found, where distance = region boundary − domain boundary, blue represents left region boundaries, and green represents right region boundaries. Thus, a negative distance in blue and a positive distance in green denote that the region boundary extends further than the domain boundary. Bin size = 10 aa.
Figure 6. Distribution of architecture family sizes. (A) The number of sequences (20 families with more than 50 sequences not shown), and (B) the number of region families per architecture family (3 families with more than 20 regions not shown) are shown. Color denotes the number of species represented in the architecture family.

Figure 7. Events per branch. The number of generation (blue), duplication (green), loss (red), merge (cyan), and split (magenta) events along each branch are shown. The number of regions in a species is displayed at each species node, and counts are totalled across all architecture scenarios.
Figure 8. Mechanisms for generating fused and fragmented gene architectures.
Figure 9. The evolution of *GJA28694* in *D. pseudoobscura* from *CG4617* and *frizzled 4 (fz4)*. (A) The MP architecture scenario. Architectures 00821 and 05692-03769 are usually in separate genes corresponding to *CG4617* and *fz4* orthologs. However, *dpse* fuses these architectures to form 00821-03769 in the gene *GA28694*. The MP evolution infers duplications of the two regions followed by a merge along the branch leading to *dpse*. (B) A genome level view shows that the orthologs of *CG4617* and *fz4* are adjacent in all species except *dpse*. In *dpse*, the new, fused gene *GA28694* is located between *CG4617* and *fz4*. (C) The hypothesized evolution of *GA28694* in *dpse* through duplication and exon shuffling. We postulate that *GA28694* was formed by duplicating *CG4617*, then duplicating the end of *fz4* and merging it with the *CG4617* duplicate. Duplication allows *dpse* to retain the original gene functions and gain a new function, and this order of events allows the *fz4*-derived region of *GA28694* to use the regulatory mechanisms of *CG4617*. 
Figure 10. The evolution of *jingwei* (jgw) in *D. yakuba* through retrotransposition and exon shuffling of alcohol dehydrogenase (*ADH*) and yellow emperor (*ymp*). (A) The MP architecture scenario. (1) Along the branch leading the dyak, the MP evolution infers the duplication of regions 17143 and 00370 followed by their subsequent merge to form jgw. (2) Along branch leading the dmel, we see an interaction between regions 17143 and 17258 to form ymp. However, there is strong evidence that the gene pairs GE10684-GE10685 and GG12235-GG12236 are actually single gene orthologs of ymp. These gene break errors lead to an incorrectly inferred ancestral architecture for ymp in which the regions 17143 and 17258 are found in separate genes rather than fused in a single gene. (3) *ADH* consists of a single region 00370, and there are multiple copies of this region in isolated form in many genomes. Multiple cases of fusions with *ADH*-derived regions have also been found experimentally, suggesting that *ADH* may be enriched for fusion events. We find region 00370 fused to the architecture 03541-01876 in dpse, which has not been previously found in literature. It is possible that GA25237 and GA25238 are further examples of *ADH*-derived chimeric genes. (B) A genome level view of ymp reveals gene break errors in the ymp orthologs. This is supported by exon structure and genome alignment and partially supported by EST evidence: ymp in dmel has multiple full ESTs (e.g. ESTs span entire gene), and GE10684 and GE10685 in dyak have multiple spanning ESTs, but none of GG12235 or GG12236 in dere nor GF17267 in dana have ESTs. *ADH* is found on a different chromosome (scaffold). (C) The hypothesized evolution of jgw in dyak. ymp is first duplicated to create a second copy yande [3], then *ADH* is retrotransposed between the third and fourth exons of yande followed by degeneration of the yande exons found after the insertion point. Exons in gray represent exons that are not part of the longest transcript.
Figure 11. The evolution of GH23361 in D. grimshawi through duplication-degeneration of rhea. (A) The MP architecture scenario. Most species have the region 09411 and 04568 fused in a single gene rhea. However, dgri has the two regions in separate genes, with the rhea ortholog containing region 09411 and the GH22519 gene containing region 04568. The MP evolution infers a split along the branch leading to dgri. Note that there is a second gene with region 09411 in the (dmel,(dyak,dere)) ancestor, which is caused by the region tree grouping dmel and dere together. This results in spurious duplication, loss, and split events being called within the melanogaster subgroup. (B) A genome level view shows that rhea and GH23361 in dgri are found on two scaffolds that alternately contain orthologs to the other eight genomes. (C) The hypothesized evolution of rhea and GH23361 in dgri through segmental duplication followed by differential degeneration. Instead of losing the entire rhea gene in one of the duplicates, rhea undergoes alternative region loss, with each copy retaining one region of the original rhea gene. This results in two genes that appear fused in the other species and fragmented in dgri.
Tables

Table 1. EST and mRNA-seq evidence indicates low rate of erroneously fragmented gene calls.

<table>
<thead>
<tr>
<th>species</th>
<th># genes</th>
<th># genes w/ EST (mRNA-seq)</th>
<th># gene pairs</th>
<th># gene pairs w/ EST (mRNA-seq)</th>
<th># gene pairs w/ spanning EST (mRNA-seq)</th>
<th>error rate (%) of EST (mRNA-seq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmel</td>
<td>14,080</td>
<td>12,640 (12,673)</td>
<td>14,052</td>
<td>11,645 (11,895)</td>
<td>78 (35)</td>
<td>0.67 (0.29)</td>
</tr>
<tr>
<td>dyak</td>
<td>16,077</td>
<td>1,618</td>
<td>15,335</td>
<td>222</td>
<td>4</td>
<td>1.80</td>
</tr>
<tr>
<td>dere</td>
<td>15,044</td>
<td>4,459</td>
<td>14,780</td>
<td>1,556</td>
<td>13</td>
<td>0.84</td>
</tr>
<tr>
<td>dana</td>
<td>15,069</td>
<td>5,022</td>
<td>14,680</td>
<td>1,864</td>
<td>24</td>
<td>1.29</td>
</tr>
<tr>
<td>dpse</td>
<td>16,099</td>
<td>2,851 (13,721)</td>
<td>15,156</td>
<td>699 (12,092)</td>
<td>13 (70)</td>
<td>1.86 (0.58)</td>
</tr>
<tr>
<td>dwil</td>
<td>15,512</td>
<td>4,699</td>
<td>14,442</td>
<td>1,792</td>
<td>17</td>
<td>0.95</td>
</tr>
<tr>
<td>dmoj</td>
<td>14,594</td>
<td>5,042</td>
<td>14,216</td>
<td>2,052</td>
<td>23</td>
<td>1.12</td>
</tr>
<tr>
<td>dvir</td>
<td>14,491</td>
<td>5,196</td>
<td>13,794</td>
<td>2,133</td>
<td>18</td>
<td>0.84</td>
</tr>
<tr>
<td>total</td>
<td>135,948</td>
<td>46,437 (39,429)</td>
<td>175,882</td>
<td>28,376 (36,110)</td>
<td>262 (187)</td>
<td>0.92 (0.52)</td>
</tr>
</tbody>
</table>

For each species, we find the number of genes with EST evidence, the number of neighboring gene pairs, the number of neighboring gene pairs in which both genes have EST evidence, and the number of neighboring gene pairs in which both genes have EST evidence and there exists at least one EST that spans both genes. Error rates are defined as the number of gene pairs with spanning EST over the number of gene pairs with EST evidence. Similar results for mRNA-seq evidence are provided in parentheses.

Table 2. GO enrichment in merge/split architecture families ranked by fold.

<table>
<thead>
<tr>
<th>rank</th>
<th>GO ID</th>
<th>GO term</th>
<th>k</th>
<th>m</th>
<th>fold</th>
<th>p-value (^4^)</th>
<th>p-value (^5^)</th>
<th>q-value (^6^)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GO:0009653</td>
<td>anatomical structure morphogenesis</td>
<td>426</td>
<td>1100</td>
<td>1.36</td>
<td>1.61 \times 10^{-14}</td>
<td>2.13 \times 10^{-7}</td>
<td>1.08 \times 10^{-4}</td>
</tr>
<tr>
<td>2</td>
<td>GO:0048731</td>
<td>system development</td>
<td>499</td>
<td>1304</td>
<td>1.34</td>
<td>8.02 \times 10^{-16}</td>
<td>2.34 \times 10^{-8}</td>
<td>1.36 \times 10^{-5}</td>
</tr>
<tr>
<td>3</td>
<td>GO:0048856</td>
<td>anatomical structure development</td>
<td>557</td>
<td>1465</td>
<td>1.34</td>
<td>5.44 \times 10^{-17}</td>
<td>8.18 \times 10^{-9}</td>
<td>5.53 \times 10^{-6}</td>
</tr>
<tr>
<td>4</td>
<td>GO:0007275</td>
<td>multicellular organismal development</td>
<td>588</td>
<td>1554</td>
<td>1.33</td>
<td>1.97 \times 10^{-17}</td>
<td>3.37 \times 10^{-9}</td>
<td>3.42 \times 10^{-6}</td>
</tr>
<tr>
<td>5</td>
<td>GO:0032502</td>
<td>developmental process</td>
<td>640</td>
<td>1709</td>
<td>1.32</td>
<td>7.95 \times 10^{-18}</td>
<td>3.03 \times 10^{-9}</td>
<td>3.42 \times 10^{-6}</td>
</tr>
<tr>
<td>6</td>
<td>GO:0032501</td>
<td>multicellular organismal process</td>
<td>711</td>
<td>1903</td>
<td>1.31</td>
<td>1.34 \times 10^{-19}</td>
<td>4.23 \times 10^{-10}</td>
<td>8.58 \times 10^{-7}</td>
</tr>
<tr>
<td>7</td>
<td>GO:0009987</td>
<td>cellular process</td>
<td>804</td>
<td>2218</td>
<td>1.27</td>
<td>3.45 \times 10^{-18}</td>
<td>5.56 \times 10^{-9}</td>
<td>4.51 \times 10^{-6}</td>
</tr>
<tr>
<td>8</td>
<td>GO:0008150</td>
<td>biological process</td>
<td>1100</td>
<td>3101</td>
<td>1.25</td>
<td>5.79 \times 10^{-22}</td>
<td>1.33 \times 10^{-10}</td>
<td>5.42 \times 10^{-7}</td>
</tr>
</tbody>
</table>

We determine the probability of obtaining at least \( k \) annotated families for a given GO term among a dataset of size \( n \), using a reference dataset containing \( m \) such annotated families out of \( N \) families. Here, \( n = 3554 \) and \( N = 14,418 \). \(^4^\) P-values using the hypergeometric test, \(^5^\) P-values corrected for length bias, \(^6^\) P-values corrected for length bias and multiple hypothesis testing (FDR). (XXX: cutoff table before biological process?)
### Table 3. Events across architecture scenarios.

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>D</th>
<th>L</th>
<th>M</th>
<th>M*</th>
<th>S</th>
<th>S*</th>
<th>D/L</th>
<th>M/S</th>
<th>M*/S*</th>
</tr>
</thead>
<tbody>
<tr>
<td>full</td>
<td># events</td>
<td>2109</td>
<td>4302</td>
<td>9873</td>
<td>4876</td>
<td>2952</td>
<td>5659</td>
<td>559</td>
<td>1:2.29</td>
<td>0.86:1</td>
</tr>
<tr>
<td># scenarios</td>
<td>1520</td>
<td>1775</td>
<td>2961</td>
<td>2242</td>
<td>955</td>
<td>2880</td>
<td>257</td>
<td>1:1.67</td>
<td>0.78:1</td>
<td>3.71:1</td>
</tr>
<tr>
<td>% scenarios</td>
<td>39.2</td>
<td>45.7</td>
<td>76.3</td>
<td>57.8</td>
<td>24.6</td>
<td>74.2</td>
<td>6.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>conserved</td>
<td># events</td>
<td>1279</td>
<td>1426</td>
<td>5763</td>
<td>2567</td>
<td>1509</td>
<td>2880</td>
<td>235</td>
<td>1:4.04</td>
<td>0.89:1</td>
</tr>
<tr>
<td># scenarios</td>
<td>1015</td>
<td>940</td>
<td>1954</td>
<td>1374</td>
<td>529</td>
<td>1747</td>
<td>81</td>
<td>1:2.08</td>
<td>0.79:1</td>
<td>6.53:1</td>
</tr>
<tr>
<td>% scenarios</td>
<td>40.7</td>
<td>37.7</td>
<td>78.4</td>
<td>55.1</td>
<td>21.2</td>
<td>70.1</td>
<td>3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The total number of events across all architecture scenarios, and the number of scenarios with at least one branch having the event type, are provided. The full dataset consists of all 3882 reconstructed architecture scenarios while the conserved dataset consists of 2506 reconstructed architecture scenarios with limited genome annotation errors. M* and S* represent simple merges and splits, e.g. merges unaccompanied by generation or duplication events, and splits unaccompanied by loss or duplication events.

### Table 4. Merge/split events tend to retain original architectures.

<table>
<thead>
<tr>
<th>MERGES</th>
<th>all</th>
<th>w/o generation</th>
<th>w/ generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of events</td>
<td>4876</td>
<td>3623 (74.3%)</td>
<td>1253 (25.7%)</td>
</tr>
<tr>
<td>retained at least one split architecture</td>
<td>4512 (92.5%)</td>
<td>3437 (94.9%)</td>
<td>1075 (85.8%)</td>
</tr>
<tr>
<td>retained both split architectures</td>
<td>2688 (55.1%)</td>
<td>2688 (74.2%)</td>
<td>n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPLITS</th>
<th>all</th>
<th>w/o loss</th>
<th>w/ loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of events</td>
<td>5659</td>
<td>2683 (47.4%)</td>
<td>2976 (52.6%)</td>
</tr>
<tr>
<td>retained merged architecture</td>
<td>1943 (34.3%)</td>
<td>1844 (68.7%)</td>
<td>99 (3.3%)</td>
</tr>
</tbody>
</table>

The first row of each table provides the total number of merges/split events, as well as whether these events are merges with a newly generated region (e.g. A → AB) or splits that also lose a split region (e.g. AB → A). Percentages are out of the total number of merge/split events. We also determine the number of merges that retain at least one split architecture (e.g. A,B → A,AB), the number of merges that retain both split architectures (e.g. A,B → A,B,AB), and the number of splits that retain the merged architecture (e.g. AB → AB,A,B), with percentages out of the number of events in the top row.