Reducing the impact of domain rearrangement on sequence alignment and phylogeny reconstruction

Sumaira Zaman[1] and Mukul S. Bansal[1,2]

[1] Department of Computer Science & Engineering, University of Connecticut, Storrs, CT 06269, USA
[2] Institute for Systems Genomics, University of Connecticut, Storrs, CT 06269, USA
sumaira.zaman@uconn.edu, mukul.bansal@uconn.edu

Abstract. Existing computational approaches for studying gene family evolution generally do not account for domain rearrangement within gene families. However, it is well known that protein domain architectures often differ between genes belonging to the same gene family. In particular, domain shuffling can lead to out-of-order domains which, unless explicitly accounted for, can significantly impact even the most fundamental of tasks such as multiple sequence alignment and phylogeny inference.
In this work, we make progress towards addressing this important but often overlooked problem. Specifically, we (i) demonstrate the impact of protein domain shuffling and rearrangement on multiple sequence alignment and gene tree reconstruction accuracy, (ii) propose two new computational methods for *correcting* gene sequences and alignments for improved gene tree reconstruction accuracy and evaluate them using realistically simulated datasets, and (iii) assess the potential impact of our new methods and of two existing approaches, MDAT and ProDA, in practice by applying them to biological gene families. We find that the methods work very well on simulated data but that performance of all methods is mixed, and often complementary, on real biological data, with different methods helping improve different subsets of gene families.

1 Introduction

Protein domains, or just *domains* for short, are independently folding structural and/or functional units that recur across multiple protein coding gene families [4]. Domains can be viewed as recurrent building blocks of proteins and are known to play an important role in the function and evolution of many gene families [20, 28, 29]. In fact, it is estimated that the majority of protein coding genes in eukaryotes and almost half of protein coding genes in prokaryotes contain at least one domain [10, 12]. Known domain sequences can be clustered into different domain families and many thousands of distinct domain families have already been identified [5].

As a gene evolves, one or more of its domains can get duplicated or be lost, and new domains can be acquired from other genes. The resulting gain and loss of domains during gene family evolution can lead to genes from the same gene family having different domain contents and architectures (i.e., sequential orderings). This is illustrated in Figure 1. Such changes in domain content and architecture through domain shuffling

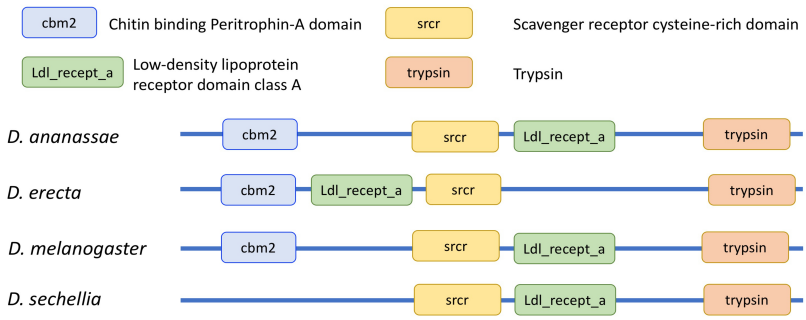

Fig. 1. Different domain architectures within a gene family. The four depicted fly proteins belong the same gene family but show different domain architectures (orderings). In particular, the order of "srcr" and "ldl_recept_a" domains appears to be inverted between *D. erecta* and *D. ananassae*. Also observe that the gene from *D. sechellia* does not have the "cbm2" domain. Note that the figure does not depict the exact location or length of any domain and only shows domain orderings.

are believed to be key drivers of protein evolution and proteome complexity [6]. As a result, mechanisms of domain shuffling and domain architecture evolution have been extensively studied in the literature [2, 7, 8, 11, 17, 25].

A frequent outcome of domain content and architecture changes within gene families is that genes belonging to the same gene family can have incompatible domain orders. For example, a gene in some gene family may have two domains A and B (from different domain families) in the order $\langle A, B\rangle$, while a different gene from the same gene family may have those domains in the order $\langle B, A\rangle$. This could occur, for example, if there is a tandem duplication of domains A and B, resulting in domain order $\langle A, B, A, B\rangle$, followed by losses of the first and last domains, resulting in the order $\langle B, A\rangle$. Such domain rearrangements, unless explicitly accounted for, can significantly impact even the most fundamental of tasks such as multiple sequence alignment and phylogeny inference. Yet, traditional approaches for computing multiple sequence alignments (MSAs) and reconstructing gene trees do not account for domain rearrangement within gene families. This is because traditional MSA algorithms perform a linear alignment of the given sequences, assuming that any variation in gene sequences is a result of point mutations or indels [1]. Domain rearrangements can violate this assumption, directly affecting the quality of the resulting MSA and of any gene trees inferred using that MSA.

Previous work. To the best of our knowledge, only three multiple sequence alignment methods, ABA [23], ProDA [22], and MDAT [13], currently exist that explicitly take domain contents and architectures into account. ABA represents a sequence alignment as a directed (possibly cyclic) graph [23], which allows for domain architecture changes and rearrangements to be detected and taken into account when analyzing evolutionary relationships between the aligned sequences. However, to our knowledge, ABA

does not compute a global multiple sequence alignment, as needed for gene tree reconstruction, and the ABA software is no longer available. ABA was also shown to have poor residue level accuracy when applied to gene families with rearranged, out-of-order domains [22]. ProDA [22] takes as input a set of unaligned sequences, uses local alignment and clustering to identify all homologous regions appearing in one or more sequences, and outputs a collection of local multiple alignments for the identified homologous regions. ProDA was shown to work well at detecting conserved domain boundaries and clustering domain segments, and at recovering known domain organizations [22]. ProDA can detect local protein homology and construct local multiple alignments, but it cannot be directly used to obtain a global alignment when the input gene sequences contain multiple domain copies from any domain family. The more recent method MDAT [13] seeks to compute more accurate MSAs by computing multiple domain alignments and restricting the global alignment such that domains from different families cannot align to each other. A limitation of MDAT is that it respects the linear arrangement of domains within each input sequence and cannot correct for rearranged, out-of-order domains. Importantly, despite the development of these previous methods, the impact of domain rearrangement on MSAs and subsequent gene tree reconstruction has not been systematically evaluated and remains largely unknown.

Our contribution. In this work, we propose two new, easy-to-apply computational methods to mitigate the impact of rearranged, out-of-order domains on gene tree reconstruction. We also carefully assess the impact of the new and previous methods on real biological data. Specifically, we first use simulated gene families, modeled after real fly gene families, to assess the impact of domain shuffling and rearrangement on MSA and gene tree reconstruction accuracy. Second, we propose two new computational approaches, referred to as *Door-S* and *Door-A* (where *Door* is short for "domain organizer"), for *correcting* gene sequences and alignments for improved gene tree reconstruction accuracy. The key idea behind our two methods is to identify known domains within the input gene sequences and then reorganize the domains to remove any domain ordering incompatibilities between the different gene sequences. This allows for an improved MSA inference for that gene family, leading to improved gene tree reconstruction. Essentially, our methods leverage the fact that standard phylogeny inference algorithms assume that sites evolve independently of each other and treat each column (site) of an MSA independently. Thus, homologous sites within gene sequences can be rearranged (together) without affecting phylogeny inference. Third, we demonstrate the impact of applying *Door-S* and *Door-A* on realistically simulated gene families. And finally, we carefully evaluate the applicability and impact of *Door-S*, *Door-A*, and the previous methods MDAT and ProDA, on biological gene families from 12 fly species. We find that the new methods result in an almost 70% average reduction in gene tree reconstruction error for the simulated gene families. However, we find that the performance of all methods is mixed when applied to the biological gene families, with the best performing methods resulting in significantly improved gene tree reconstruction for about a quarter of the gene families but showing either comparable or worse reconstruction accuracy for the other gene families. Interestingly, the performance of the different methods on biological data is often complementary, with different methods helping im-

prove different subsets of gene families. Scripts implementing *Door-S* and *Door-A* are freely available from `https://github.com/suz11001/Door/tree/main`.

2 Description of Methods

2.1 Proposed Methods: *Door-S* and *Door-A*

Both *Door-S* and *Door-A* seek to identify and reorganize domains within each input gene sequence to enable and improve the alignment of homologous regions in the final global MSA. The main steps in the *Door-S* and *Door-A* methods are as follows:

1. Identification of domain families present within the gene family.
2. Identification of domain sequence boundaries and non-domain regions within each gene sequence.
3. Ordering of non-domain regions and domain families for each gene.
4. Ordering of domains copies from same domain family within each gene.
5. Computation of final global MSA.

Door-S and *Door-A* differ only in their implementation of Step 5 above. Specifically, *Door-S* uses a traditional multiple sequence aligner, such as MUSCLE [9], to globally align the reorganised gene sequences, while *Door-A* separately aligns the different domain families and non-domain regions and concatenates these alignments to create a global concatenated alignment for the gene family. Figure 2 illustrates the shared and individual steps of *Door-S* and *Door-A*. We elaborate on these steps below.

1. Identification of domain families present: Domain families present within gene sequences can be identified using protein domain databases or tools such as Pfam [19], SMART [26], PANTHER [18] or InterPro [21]. For our biological dataset from 12 fly species, we used UniProt gene IDs to determine their protein domain constituents from the Pfam A database.

2. Identification of domain sequence boundaries and non-domain regions: Domain annotations are imperfect and the domain sequences found in PFAM or any other domain database may not be an exact match to the domain sequence present in the gene. We therefore align each annotated domain sequence back to the gene and extract the precise genic region where the annotated domain aligns. In case multiple annotated domains from different domain families overlap in the genic space, we duplicate the regions of alignment where the domains overlap. Once all the domain regions of the gene have been identified, these domain regions are removed from the gene sequences and are placed as domain sequences as part of their respective domain families.

3. Ordering of non-domain regions and domain families: The domain and non-domain (genic) regions within each gene sequence of a gene family are ordered such that the genic regions appear first, followed by the domain family sequences in a fixed order. This ensures that the ordering of domain family sequences remains consistent between all genes belonging to the same gene family. This is illustrated in the top half of Figure 2.

4. Ordering of domain copies from same domain family: If a gene sequence contains multiple domain copies from the same domain family then we place these copies

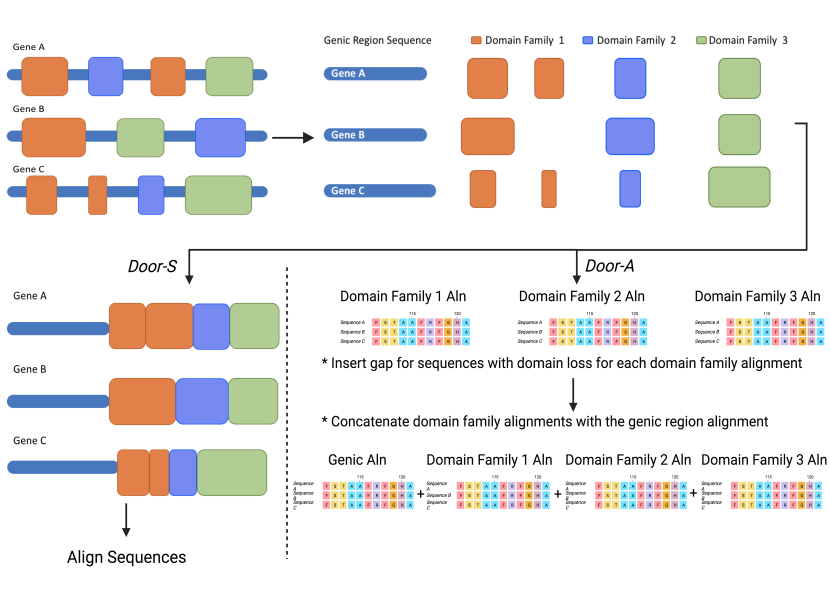

Fig. 2. Overview of *Door-S* and *Door-A*. The key difference between the two methods is that *Door-S* first concatenates all genic (non-domain) and domain sequences in a consistent order across all gene sequences and then performs global sequence alignment of the resulting, reordered gene sequences as the final step (see bottom left of figure). In contrast, *Door-A* first separately aligns the genic sequences and sequences from each domain family and concatenates the resulting alignments, introducing gaps for sequences with domain losses, to obtain the final global alignment (see bottom right of figure).

contiguously in the same order as in the original gene sequence. A different approach was used for the simulated gene families since we did not explicitly simulate domain orderings in the gene sequences; specifically, for each domain family, we choose a reference gene g_{ref} with the most number of domain copies of that domain family and greedily align the domain copies in the other genes from that gene family to the most similar domain copy in g_{ref}.

5. Computation of final global MSA. *Door-S* and *Door-A* take different approaches for this final step, as illustrated in the bottom half of Figure 2. ***Door-S*** concatenates the reordered genic and domain region sequences within each gene to create a reordered version of each original gene sequence. These reordered gene sequences, all from the same gene family, are then globally aligned using a standard global aligner. In this work, we used MUSCLE v. 3.8.31 [9] with default parameters to compute all alignments. Instead of first concatenating the reordered genic and domain regions and then aligning the resulting concatenated sequences, ***Door-A*** first aligns the genic regions and each domain family separately, and then concatenates the resulting alignments to obtain a

final global alignment of the reordered gene sequences. As part of this process, to ensure a well-formed final global alignment, gaps are artificially introduced if a domain family is completely absent from a gene sequence.

2.2 Existing methods: MDAT and ProDA

We also evaluate the two related previous methods ProDA [22] and MDAT [13]. Even though MDAT cannot correct for rearranged, out-of-order domains, it can be directly used to compute a global sequence alignment for multi-domain gene families. MDAT relies on protein domain annotations generated using a specific version (version 27) of the Pfam domain database and uses this annotation to restrict the global alignment, ensuring that domains belonging to different domain families cannot be aligned together.

ProDA takes as input a set of unaligned sequences and uses local alignment and clustering to identify all homologous regions appearing in one or more input sequences. It outputs a collection of local multiple alignments for the identified homologous regions. However, ProDA does not compute a global sequence alignment and cannot be directly used to compute one based on the output alignment blocks. For instance, some sequence segments, or even entire genes, do not appear in any output alignment, and each alignment block can contain multiple homologous regions from the same gene sequence. Nonetheless, ProDA's effectiveness at identifying regions of local homology can be leveraged to identify and correct for out-of-order domains or other regions. Accordingly, to apply ProDA to this problem, we use a scheme similar to that used for *Door-A* to compute global sequence alignments from ProDA's output: First, we modify each block of homologous sequences by identifying domain copies from the same gene sequence and arrange them linearly according to their ordering in the gene sequence. This step is similar to Step 3 of *Door-S* and *Door-A*. Second, we compute a sequence alignment (using MUSCLE) for each modified block of homologous sequences (similar to Step 5 of *Door-A*). Third, we add back the genes not represented in the resulting alignment by introducing gaps in the alignment for that gene. Finally, we concatenate the alignment for each blocks of homologous sequences to obtain an overall global alignment for the gene sequences of that gene family.

ProDA also has an input parameter which controls for the minimum size of a homologous sequence block. In our evaluation of ProDA, we used two settings for this parameter; one in which the minimum block size is set to 50 amino acids (aa), and another in which the parameter value is set to the length of the shortest Pfam domain sequence found in that gene family. We refer to these two executions as ProDA_50 and ProDA, respectively.

3 Dataset Description and Experimental Setup

3.1 Simulated dataset

We first used simulated gene family sequences, with known ground truth, to assess the impact of domain rearrangements on gene tree reconstruction and demonstrate the impact of *Door-S* and *Door-A*. To enhance the biological realism of this simulated dataset,

we selected key parameter values, such as for gene length, average number of domain families per gene famiy, average domain length, and average number of domain rearrangements, based on a real datset from 12 fly species (described later in this section). Specifically, starting with a biological dataset of 2307 multi-domain gene families from 12 fly species (see Section 3.2), we first identified 198 gene families with plausible out-of-order domains using the simple procedure described in Section 3.2. Essentially, this procedure identifies those gene families which contain at least one pair of genes whose domain orderings are incompatible with each other. For these 198 gene families, we find a median genic (not counting domain sequences) length of 452 aa, median domain sequence length of 78 aa, median of 3.6 *unique* domain families per gene family, and median of 1 for the number of unique out-of-order domain-family pairs present. We also estimated the probability of any given gene sequence having out-of-order domains. This probability depended on gene family size y, and was estimated to be 0.45, 0.27, 0.24, 0.15, 0.34, and 0.22 for $y \leq 10$, $10 < y \leq 25$, $25 < y \leq 50$, $50 < y \leq 75$, $75 < y \leq 100$, and $y > 100$, respectively.

Simulating gene trees and domain trees. Based on these parameter estimates, we used the phylogenetic simulation framework SaGePhy [14] to generate 100 gene families and their corresponding domain families. First, we simulated 100 species trees with SaGePhy using a birth-death model with birth and death rate of 5 and 2, respectively, and height 1. A gene tree was then evolved inside each species tree under a duplication-loss model with gene duplication and gene loss rates of 0.3 each. Finally, we evolved 3 domain trees inside each gene tree with domain duplication and domain loss rates of 0.3 each. This yielded gene families with similar domain characteristics as the biological dataset.

Simulating sequence data. We then used SaGePhy to simulate protein sequences along both the gene and the three domain trees under the LG amino acid substitution model [15] and appended together (in a predetermined order) the genic and domain sequences belonging to the same gene. Hence, each gene consists of a genic (non-domain) sequence and a variable number of domain sequences from one or more domain families. Each genic sequence is 450 aa long and each domain sequence is 100 aa long, so that each full gene sequence has length 450 aa or more depending on the number of domain sequences present in it.

Introducing rearrangements. After creating these baseline sequences for each gene in the gene family, we introduce domain rearrangement in a randomly chosen subset of the gene sequences based on the probabilities previously estimated from the biological dataset. We follow a conservative procedure for introducing domain rearrangements where we only make one rearrangement (exchange the positions of a single pair of domains) in each selected gene sequence. In most cases, we only exchange two neighboring domains. For example, if the simulated gene sequence shows the domain ordering $([A1, A2, A3], [B1, B2], [C1])$, where A, B, and C represent the three domain families, then, in most cases, we only exchange either $A3$ with $B1$ or $B2$ with $C1$, thereby creating exactly one pair of out-of-order domain sequences in that gene sequence. Based on observations in the biological dataset, we also sometimes perform rearrangements so as not to disrupt the tandem ordering of domain copies. For example, if the simulated

gene sequence shows domain ordering $([A1, A2], [B1], [C1])$, then we rearrange the sequence to $([B1], [A1, A2], [C1])$ with a small probability based on biological data.

3.2 Biological dataset

As our real biological dataset, we used the 12-flies dataset assembled by Li et al. [16] in their study of protein domain evolution. This dataset consists of 7165 gene families in which at least one gene has at least one Pfam A domain. Of these 7165 gene families, 2307 gene families contain domains from at least two domain families. Among these 2307 gene families, we identified 198 as having plausible out-of-order domains and our experimental results are based on these 198 gene families.

The 198 gene families with plausible out-of-order domains were identified as follows: We first represent each gene sequence by its ordering of domains. For example, a gene sequence consisting of 8 distinct domains from 4 different domain families A, B, C and D would be represented as follows, based on the specific ordering of the 8 domain sequences: [(A),(A),(B),(B),(C),(D),(C),(B)]. For simplicity and to avoid possible overcounting of out-of-order domains, we then condense the above representation by merging together contiguous domains from the same domain family. Thus, the representation for the above gene would be condensed to [(A),(B),(C),(D),(C),(B)]. We then consider the condensed representations of each pair of gene sequences from the gene family and check if that pair of genes has incompatible domain orders. More precisely, we check if there exists a domain family pair $\{$X,Y$\}$ such that this pair occurs only in the order $\langle X, Y \rangle$ in one of the gene sequences and only in the order $\langle Y, X \rangle$ in the other gene sequence. If we find any pair of gene sequences to have incompatible domain orders then we flag that gene family as plausibly having out of order domains.

3.3 Evaluation of results

The most commonly used accuracy metric for multiple sequence alignments is the sum-of-pairs (SP) score. SP scores are computed by comparing every pair of amino acids in an aligned column to assign an alignment quality score to that column, and then summing up these scores across all columns in the alignment. The higher the total score, the better the quality of the alignment. However, this scoring scheme is only appropriate when the sequences being aligned are actually alignable. For sequences with out-of-order domains, the SP score can yield misleading results and need not be correlated with gene tree reconstruction accuracy. We will see a clear example of this in the next section. Consequently, we assess the impact of out-of-order domains and of the different correction methods based on reconstructed gene tree accuracy. We measure gene tree accuracy by comparing each reconstructed gene tree against the corresponding ground truth gene tree using the standard Robinson-Fould's metric [24]. Specifically, we count the number of splits present in only one of the two trees being compared (the reconstructed vs the true gene tree). We refer to the resulting number as the RF-score, with a lower RF-score implying greater gene tree reconstruction accuracy. Note that the reported RF-scores count unique splits of both trees (i.e., we do not divide the computed score by 2).

Since ground truth gene trees are only available for the simulated dataset, gene tree accuracy cannot be directly measured for the biological dataset. To overcome this challenge, we use the reconciliation cost (specifically the duplication-loss reconciliation cost) of each reconstructed gene tree against the known 12-flies species tree as a proxy for gene tree accuracy. We compute this reconciliation cost under a parsimony framework [3] using a loss cost of 1 and a duplication cost of 2. We refer to the resulting cost as the DL-score. In section 4.1, using the simulated datasets, we show that the DL-score generally increases or decreases in line with the RF-score (i.e., greater gene tree error results in a higher DL-score), thereby justifying its use as a proxy for the RF-score.

3.4 Gene tree reconstruction

For each simulated gene family, we reconstruct four gene trees based on the following four gene family alignments: (i) an alignment (using MUSCLE [9]) of the simulated baseline sequences with no domain rearrangement, (ii) an alignment (using MUSCLE) of the rearranged gene sequences, (iii) the alignment produced by applying *Door-S* to the rearranged sequences, (iv) and the alignment produced by applying *Door-A* to the rearranged sequences. Thus, the first tree represents the baseline scenario when there is no domain rearrangement in the gene sequences and captures baseline alignment and gene tree reconstruction error. The second tree represents the scenario when domain rearrangement is present but is not accounted for in the gene family alignment. The third and fourth trees represent the scenarios when domain rearrangement is present and has been corrected for using *Door-S* and *Door-A*, respectively. All simulated dataset gene trees were reconstructed using RAxML v8.2.11 [27] with thorough search settings (-f a -N 100) and under the same model (PROTGAMMAILG) used for the simulation.

For the biological dataset, we reconstruct six gene trees for each of the 198 gene families. These six gene trees correspond to the original (uncorrected) MUSCLE alignment and the corrected alignments obtained by applying *Door-S*, *Door-A*, MDAT, ProDA, and ProDA_50. All biological dataset gene trees were reconstructed using RAxML v8.2.11 [27] with thorough search settings under the PROTGAMMAAUTO model.

4 Results

4.1 Simulated dataset results

Table 1 summarises our results for the simulated dataset. As the table shows, introducing domain rearrangements in the gene sequences leads to a dramatic worsening of gene tree reconstruction accuracy, with the average RF-score increasing from 9 for the baseline sequences without rearrangement to 43 for the aligned rearranged sequences. The table also shows the drastic improvement in gene tree accuracy obtained after correcting the rearranged sequences using *Door-S* and *Door-A*. Specifically, the RF-score decreases from 43 to only 14 and 13, respectively, after *Door-S* and *Door-A* are applied. Overall, among the 100 simulated gene families in this dataset, *Door-S* resulted in an improved RF-score for 95 gene families and *Door-A* for 97 gene families. These results show that both *Door-S* and *Door-A* are highly effective at correcting MSAs for improved gene tree reconstruction, with *Door-A* slightly outperforming *Door-S*.

Table 1. Gene tree reconstruction accuracy using different alignment types for the simulated gene families. Accuracy is shown in terms of RF-scores, averaged across the 100 gene families in the simulated dataset. Corresponding average DL-scores and SP-scores are also shown. Lower values are better for RF-score and DL-score, while higher values are better for SP-score. Observe that DL-scores are well-aligned with RF-scores, but that SP-scores are not, with the *Door-A* corrected alignment showing the worst (lowest) SP-score among all four alignment types.

Alignment type	**RF-score**	DL-score	SP-score
Baseline sequences alignment (no rearrangement)	9	40	626
Rearranged sequences alignment	43	117	576
Door-S corrected alignment	14	51.25	635
Door-A corrected alignment	13	48	550

Relationship between RF-score and DL-score. Table 1 also shows average DL-scores for the gene trees reconstructed using the four alignment types. As the table shows, these DL-scores are highly correlated with corresponding RF-scores, increasing and decreasing by similar degrees as the RF-scores. Overall, we observed that application of *Door-S* and *Door-A* resulted in improved (decreased) DL-score in 95 of the 100 gene families. These observations justify the use of DL-score as a proxy for gene tree reconstruction error for the biological dataset where true gene trees are unknown.

Inapplicability of SP-score. We also computed SP-scores for the *Door-S* and *Door-A* alignments and compared them to SP-scores for the rearranged sequence alignments (Table 1). Based on the drastic improvement in gene tree accuracy enabled by *Door-S* and *Door-A*, one would expect the *Door-S* and *Door-A* alignments to show much better (higher) SP-scores. While all *Door-S* alignments do show an improvement, we found that only 11 of the 100 *Door-A* alignments had an improved SP-score compared to rearranged sequence alignments. In other words, 89% of the *Door-A* alignments actually had worse SP-scores than the rearranged sequence alignments. The finding that *Door-A* alignments have worse SP-scores than *Door-S* alignments is not is not surprising; specifically, *Door-A* alignments are composed of concatenated alignments of smaller sequence blocks and are therefore more "restricted" compared to the *Door-S* alignments where the aligner has greater opportunity to improve the SP-score by aligning matching nucleotides (or amino acids) across domain boundaries. These results demonstrate how SP-scores need not be correlated with alignment quality or gene tree reconstruction accuracy in the presence of domain rearrangement.

Note that we did not apply MDAT and ProDA to the simulated dataset. ProDA (or ProDA_50) could not offer any improvement over *Door-A* since exact domain families and domain sequence boundaries are already known for the simulated dataset. MDAT could not be applied since it requires specifically formatted PFam annotations which are unavailable for the simulated data.

4.2 Biological dataset results

We applied all five methods, *Door-S*, *Door-A*, MDAT, ProDA, and ProDA_50 to the 198 biological gene families and compared the accuracies of the resulting gene trees against

Table 2. Number of gene families improving or worsening, per the DL-score, when applying MDAT, ProDA, ProDA_50, *Door-S*, and *Door-A* to the 198 biological gene families.

Method	No. of Families Improved	Avg. Percent Improvement	No. of Families Worsened	Avg. Percent Worsening
MDAT	50	16.4	114	37.4
ProDA	39	19.1	120	26.3
ProDA_50	45	17	119	31.6
Door-S	42	15.4	125	36
Door-A	49	16	114	34.6

the gene trees constructed using the original (uncorrected) gene sequence alignments.[3] Since true gene trees are unavailable for the biological dataset, relative gene tree accuracies were estimated based on DL-scores, as described previously. Table 2 shows the results of this analysis. In contrast with the results on simulated datasets, we observed that none of the methods could consistently improve all gene families and that the majority of gene families showed worse accuracy after the methods were applied. The best performing methods on this dataset were MDAT and *Door-A*, which both improved approximately 25% of the gene families and worsened 57.6% of the gene families.

We also observed that the different methods tended to improve different subsets of gene families; see Figure 3. As expected, the greatest overlap in improved gene families occurs for *Door-S* and *Door-A* and for ProDA and ProDA_50. When considering only the best performing method of each type, MDAT, ProDA_50, and *Door-A*, we find that they all show an improvement for 15 shared gene families (Figure 3(a)). This level of overlap is highly unlikely to occur by chance (p value < 0.0001). In fact, based on 10,000 randomization experiments, we observed an average overlap of only 2.8 gene families for the three methods. This suggests that it may be possible to predict which gene families would benefit from the application of such methods.

These results also highlight the difficulty of dealing with domain rearrangement in real biological gene families. In particular, error-prone identification of domains and domain boundaries, and inability to identify all homologous regions affected by rearrangement can all greatly impact *Door-S* and *Door-A*, as well as the other methods. The competitive performance of MDAT on these gene families also suggests that, for several of the gene families, the gene sequences may actually be linearly alignable. E.g., the seemingly incompatible domain orders $\langle A, B\rangle$ and $\langle B, A\rangle$ become linearly alignable in the presence of a third sequence with domain order $\langle A, B, A\rangle$.

5 Discussion and Conclusion

In this work, we considered the problem of out-of-order domains within gene families. We used carefully simulated gene families to demonstrate the impact of protein domain shuffling and rearrangement on multiple sequence alignment and gene tree reconstruction accuracy, proposed two new computational methods, *Door-S* and *Door-A*,

[3] ProDA and ProDA_50 could only be run successfully on 183 and 191 gene families, resp.

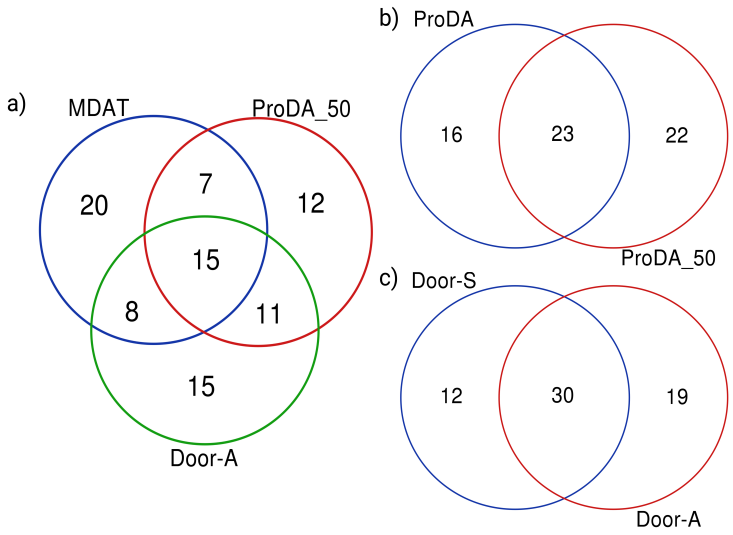

Fig. 3. Venn diagrams for gene families improved by different methods. **(a)**: Venn diagram showing intersections of improving gene families for the three primary methods, MDAT, ProDA_50, and Door-A. **(b)** and **(c)**: Venn diagrams showing intersections of improving gene families for similar methods ProDA & ProDA_50 **(b)** and *Door-S* & *Door-A* **(c)**.

for correcting gene sequences and alignments for improved gene tree reconstruction accuracy, demonstrated their drastic impact on gene tree reconstruction accuracy on the simulated dataset, and assessed the potential real-world impact of the new methods and MDAT and ProDA by applying them to biological gene families. Our findings demonstrate the significant impact that proper handling of domain rearrangements can have on gene tree reconstruction accuracy, and identify the substantial challenges that such methods must overcome to become widely applicable in practice. Notably, none of the evaluated methods could consistently improve the accuracy of reconstructed gene trees for the biological gene families.

Between *Door-S* and *Door-A*, our experimental results on both simulated and biological gene families indicate that the concatenated alignment approach implemented in *Door-A* may be slightly superior, overall, to the simpler approach implemented in *Door-S*. However, we found that there were several biological gene families that were improved by *Door-S* but not by *Door-A* (Figure 3), and further work is needed to better understand the scenarios in which one or the other method works better. Our results on the biological dataset suggest that both *Door-S* and *Door-A* could be further improved by combining protein domain annotations with the local alignment approach of ProDA to better identify out-of-order domains and other homologous regions and their boundaries. Our results also suggest that first constructing an order-preserving alignment, as done by MDAT, may help to better identify gene families with true out-of-order domains which could benefit from the reordering-based approach of *Door-S* and *Door-A*.

Finally, our results are based on using MUSCLE as the underlying sequence aligner and it could be instructive to assess the impact of using other sequence aligners to compute baseline alignments and *Door-S* and *Door-A* corrected alignments.

Funding: This work was supported in part by NSF award IIS 1553421 to MSB.

References

1. Anders Krogh, Sean Eddy, Richard M. Durbin: Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1998)
2. Baker, E.P., Sayegh, R., Kohler, K.M., Borman, W., Goodfellow, C.K., Brush, E.R., Barber, M.F.: Evolution of host-microbe cell adherence by receptor domain shuffling. Elife **11** (Jan 2022)
3. Bansal, M.S., Kellis, M., Kordi, M., Kundu, S.: RANGER-DTL 2.0: rigorous reconstruction of gene-family evolution by duplication, transfer and loss. Bioinformatics **34**(18), 3214–3216 (2018)
4. Björklund, A.K., Ekman, D., Light, S., Frey-Skött, J., Elofsson, A.: Domain rearrangements in protein evolution. J. Mol. Biol. **353**(4), 911–923 (Nov 2005)
5. Blum, M., Chang, H.Y., Chuguransky, S., et al.: The InterPro protein families and domains database: 20 years on. Nucleic Acids Research **49**(D1), D344–D354 (11 2020)
6. Choudhuri, S.: Chapter 2 - fundamentals of molecular evolution. In: Choudhuri, S. (ed.) Bioinformatics for Beginners, pp. 27–53. Academic Press, Oxford (Jan 2014)
7. Cohen-Gihon, I., Sharan, R., Nussinov, R.: Processes of fungal proteome evolution and gain of function: gene duplication and domain rearrangement. Phys. Biol. **8**(3), 035009 (Jun 2011)
8. Dohmen, E., Klasberg, S., Bornberg-Bauer, E., Perrey, S., Kemena, C.: The modular nature of protein evolution: domain rearrangement rates across eukaryotic life. BMC Evol. Biol. **20**(1), 30 (Feb 2020)
9. Edgar, R.C.: MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics **5**, 113 (Aug 2004)
10. Ekman, D., Åsa K. Björklund, Frey-Skött, J., Elofsson, A.: Multi-domain proteins in the three kingdoms of life: Orphan domains and other unassigned regions. Journal of Molecular Biology **348**(1), 231 – 243 (2005)
11. Forslund, K., Sonnhammer, E.L.L.: Evolution of protein domain architectures. In: Anisimova, M. (ed.) Evolutionary Genomics: Statistical and Computational Methods, Volume 2, pp. 187–216. Humana Press, Totowa, NJ (2012)
12. Han, J.H., Batey, S., Nickson, A.A., Teichmann, S.A., Clarke, J.: The folding and evolution of multidomain proteins. Nature Reviews Molecular Cell Biology **8**, 319–330 (2007)
13. Kemena, C., Bitard-Feildel, T., Bornberg-Bauer, E.: MDAT- aligning multiple domain arrangements. BMC Bioinformatics **16**, 19 (Jan 2015)
14. Kundu, S., Bansal, M.S.: SaGePhy: an improved phylogenetic simulation framework for gene and subgene evolution. Bioinformatics **35**(18), 3496–3498 (Feb 2019)
15. Le, S.Q., Gascuel, O.: An improved general amino acid replacement matrix. Mol. Biol. Evol. **25**(7), 1307–1320 (Jul 2008)
16. Li, L., Bansal, M.S.: An integrated reconciliation framework for domain, gene, and species level evolution. IEEE/ACM Trans. Comput. Biol. Bioinform. **16**(1), 63–76 (Jan 2019)
17. Marsh, J.A., Teichmann, S.A.: How do proteins gain new domains? Genome Biol. **11**(7), 126 (Jul 2010)
18. Mi, H., Thomas, P.: PANTHER pathway: an ontology-based pathway database coupled with data analysis tools. Methods Mol. Biol. **563**, 123–140 (2009)

19. Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G.A., Sonnhammer, E.L.L., Tosatto, S.C.E., Paladin, L., Raj, S., Richardson, L.J., Finn, R.D., Bateman, A.: Pfam: The protein families database in 2021. Nucleic Acids Res. **49**(D1), D412–D419 (Jan 2021)
20. Miyata, T., Suga, H.: Divergence pattern of animal gene families and relationship with the cambrian explosion. BioEssays **23**(11), 1018–1027 (2001)
21. Paysan-Lafosse, T., Blum, M., Chuguransky, S., et al.: Interpro in 2022. Nucleic Acids Res. **51**(D1), D418–D427 (2023)
22. Phuong, T.M., Do, C.B., Edgar, R.C., Batzoglou, S.: Multiple alignment of protein sequences with repeats and rearrangements. Nucleic Acids Res. **34**(20), 5932–5942 (Oct 2006)
23. Raphael, B., Zhi, D., Tang, H., Pevzner, P.: A novel method for multiple alignment of sequences with repeated and shuffled elements. Genome Res. **14**(11), 2336–2346 (Nov 2004)
24. Robinson, D.F., Foulds, L.R.: Comparison of phylogenetic trees. Math. Biosci. **53**(1), 131–147 (Feb 1981)
25. Sato, P.M., Yoganathan, K., Jung, J.H., Peisajovich, S.G.: The robustness of a signaling complex to domain rearrangements facilitates network evolution. PLoS Biol. **12**(12), e1002012 (Dec 2014)
26. Schultz, J., Copley, R.R., Doerks, T., Ponting, C.P., Bork, P.: SMART: a web-based tool for the study of genetically mobile domains. Nucleic Acids Res. **28**(1), 231–234 (Jan 2000)
27. Stamatakis, A.: RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics **30**(9), 1312–1313 (May 2014)
28. Tordai, H., Nagy, A., Farkas, K., Banyai, L., Patthy, L.: Modules, multidomain proteins and organismic complexity. FEBS Journal **272**(19), 5064–5078 (2005)
29. Vogel, C., Bashton, M., Kerrison, N.D., Chothia, C., Teichmann, S.A.: Structure, function and evolution of multidomain proteins. Current Opinion in Structural Biology **14**(2), 208 – 216 (2004)