– Independent Work Report Spring, 2020 –

Leveraging Sequencing Structure for Accurate Phylogenetic Analysis Of Protein Domains

William Lin
Adviser: Mona Singh, Chaitanya Aluru

Abstract

This paper describes the steps by which a new phylogenetic tree reconstruction algorithm was developed. This algorithm directly builds upon previously conducted research in designing accurate simulations for protein domain evolution, and was developed using a combination of Python packages for creating and analyzing phylogenetic trees, widely available conventional tree reconstruction algorithms, a new method for reconstructing gene tree topologies from sequence groups. Multiple phylogeny reconstructions were generated from random, realistic domain-evolution simulations, and reconstructed trees were compared against simulated trees to evaluate reconstruction accuracy. This reconstruction algorithm attempts to provide novel insight regarding some techniques which may improve the reconstruction accuracy of widely available tree reconstruction algorithms.

1. Introduction

1.1. Gene Families and Protein Domains

Gene families are sets of genes which are related ancestrally and encode functionally related proteins. Gene families form as a result of repeated duplication and diversification of highly conserved genes. Gene families have been observed to have expanded and diversified across multiple lineages, and are often considered a characteristic feature of higher-level organisms [13]. Previous analysis of gene families has indicated that gene gains and loss as a result of this diversification reflect the largest genetic differences which separate humans and their closest relatives [11]. Notable gene families include those which code for important protein domains, the structural and functional building blocks of proteins. Protein domains are conserved protein sequences with distinct structure,

function, and evolutionary history [7]. These protein domains can bind to and process ligands, often playing a central role in signaling cascades by binding to specific signaling-molecules. Protein domains have been shown to behave as independent genetic elements within genomes, and due to their relatively conserved nature, are often identifiable from their nucleotide and amino acid sequences. Analyse have shown that protein domains are found in all three domains of life (Archaea, Bacteria, and Eukarya) [18]; limited combinations of protein domains are shared between the three domains of life, indicating that domains are reused and shared during evolution and serve as important functional units in those processes [5] [22].

Protein domains often combine with one another in order to form larger, multi-domain proteins. Research suggests that domain combinations are evolutionarily conserved and that diversification of these combinations creates novel functions [17]. The diversity of these domain combinations and the evolution of the gene families for which they code has been arises in part form the evolutionary processes in higher-order organisms (e.g. humans) including gene recombinations, gene fusion and fission, alternative splicing. These domain combinations often take the form of regular structures including repeated β-sheets or solenoids [3], and are especially common in multicellular species [4]. Previous studies have indicated that protein classes with the highest incidence of repetitive sequences carry out important functions in eukaryotes and also serve to increase genetic variability amongst eukaryotes. [15].

Even amongst well-defined and conserved protein domains, variation in sequences and domain repeat length allows for the aforementioned domain repeats to both bind to a plethora of different partners and perform highly diverse functions. Many protein domains which originated in specific lineages carry different functions and and have diversified across multiple lineages, including humans and other organisms. For example, the Kruppel-associated box (or KRAB domain) is found in several vertebrate species, particularly mammals. KRAB domain repeats code for the formation of a thick layer of cross-linked proteins in keratinocytes during the formation of the skin. KRAB domains are also prevalent Cys2Hys2 Zinc Finger gene family, more generally known as zinc fingers. Zinc fingers are particularly notable as they are the most diverse and numerous class

of transcription factors found in many multicellular animals (see Figure 1), and alone account for more than half of the transcription factors in the human genome (see Figure 2).

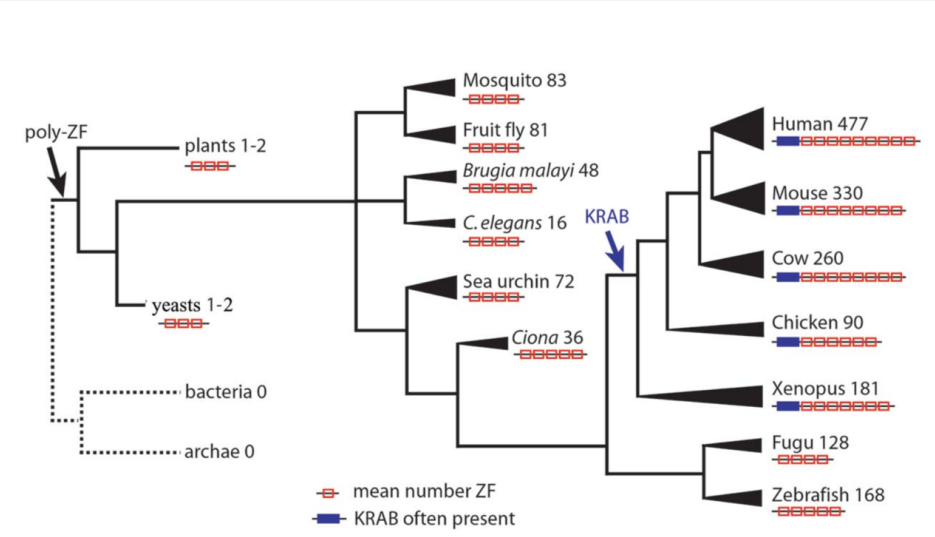

Figure 1: Diversification of zinc fingers in animals.

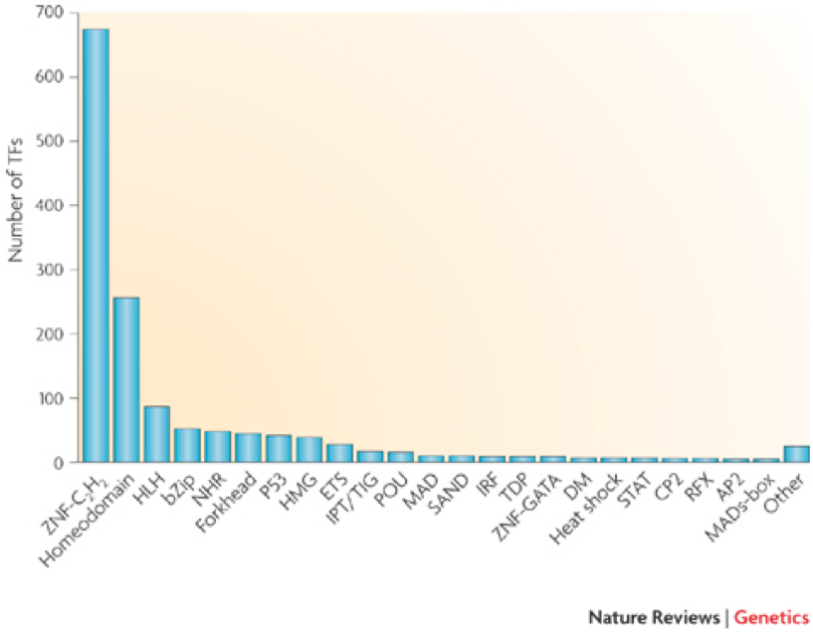

Figure 2: Zinc finger prevalence and transcription factor frequencies within humans.

Despite the fact that protein domains play an important role in a multitude of biological processes, we don't know much about their evolutionary mechanisms and how they diversified in Eukaryotes. Studies suggest that the rapid expansion of protein domain repeats evolved through internal tandem

duplications where segments are duplicated and directly inserted next to their origin [7], recombination of intron (coding) regions, or DNA "slippage" which occurs as a result of DNA hairpins [12]. Because these duplication and recombination events commonly occur during evolution [14] and protein domain-coding genes are so prevalent in humans, learning more about the evolutionary histories of these proteins and how they diversified into modern day sequence groups is of the utmost importance.

1.2. Motivation and Goal

This research project is a direct continuation of William Lin's research, "Simulating Domain-Level Evolution in Zinc Fingers", completed in Spring 2019 [2]. In the previous research project, a novel simulation for the evolution of zinc finger protein domains was developed. This simulation accurately captured two evolutionary phenomenons: 1) domain-level events, namely block duplications, losses, and speciations, and 2) sequence-level evolution of the protein domains and their surrounding linker regions. The conclusions from the analysis indicated that the simulation model was capable of generating realistic internally similar orthologous groups from a valid zinc finger protein sequence.

The goal of this research project is to develop a new algorithm capable of accurately generating phylogenetic tree topologies from sequences containing protein domains such as the simulated zinc finger sequence groups with which we previously worked. There currently exist multiple options for phylogenetic tree reconstruction algorithms, but these algorithms offer little regarding the evolutionary patterns of protein domains. Because domains are highly conserved functional units, traditional reconstruction algorithms are relatively ineffective at recreating accurate and realistic histories for them. There does not exist widely effective solution for inferring evolutionary histories for sequences which contain highly conserved protein domains. In addition, many pre-existing methods are too inefficient to run on meaningfully sized datasets (e.g. TreeFix is a brute force search algorithm searching over an exponential space and can take upwards of minutes per input sequence).

Thus, research project hopes to directly build upon the previous research, effectively "working backwards" to develop a useful software package for analyzing protein domains. Due to the nature of domain evolution, we hope to develop a reconstruction technique which tightly integrates gene and species information to address domain specific phenomena and infers gene trees which represent fully accurate evolutionary histories. By generating an accurate reconstruction, we will be able to learn more about the mechanisms by which domain-level evolution occurs.

2. Background and Related Work

2.1. Importance of Phylogeny

Phylogeny refers to the history of evolution amongst a species and its populations; phylogenetics researchers examine lines of descent and relationships between different groups of organisms in order to gain new insights regarding the species' evolutionary development and how different groups of organisms and biological features diversified over time. Analyzing the relationships between different species or gene sequences provides a much needed framework for analyzing much of modern biology; understanding phylogenetics is widely applicable to researchers in a variety of sub-fields, including but not limited to ecology, physiology, molecular biology, and in particular, genomics [20]. Phylogenetic studies are a valuable tool for genomics researchers examining patterns of evolution exhibited by many genetic features, many of which have significant functional importance in organisms. The insights garnered by phylogenetics reveals much about modern organisms which contain genes of interest as well as the full evolutionary history of those genes. For example, phylogenetic analysis of gene families coding for MADS box genes, which encode transcription factors which control diverse processes in plants, revealed a common ancestor for multiple current angiosperms [6].

One of the fundamental challenges in phylogenetics is accurately reconstructing these evolutionary histories from current groups of sequences. This challenge is particularly difficult for short sequences such as domains and genes due to the limited genetic information available.The accuracy by which we can infer the evolutionary history of a gene directly affects a variety of downstream analyses

[9]. Reconstructions of gene trees are utilized to draw inferences about adaptations, understand evolutionary events (gene loss, gene duplication, horizontal gene transfer), identify orthologs, track the evolution of functional traits, estimate species trees, and much more. However, there exist multiple complications when dealing with gene trees. The evolutionary behavior of gene sequences is complicated by horizontal gene transfers, hybridization events, duplications, losses, and reticulations, making it difficult to determine the correct gene tree topology with high confidence. Unlike sequence trees, gene tree reconstruction operates with shorter sequences, often at a single locus, and typically does not strong enough signal to recreate accurate histories.

2.2. Current Status of Phylogenetic Tree Reconstruction Algorithms

Phylogenetic tree reconstruction algorithms can be divided into four general categories based on their required inputs, how many topologies are generated, and any additional information the method provides. Examining phylogenetic tree reconstruction algorithms provide insight about available methods and their flaws.

Neighbor-Joining: Distance-based methods like estimate the average number of changes per site since two sequences deviated. Neighbor joining methods estimate pair-wise distances between sequences, and transform observed percent differences into a projected number of nucleotide substitutions using a model of molecular evolution. Neighbor-joining algorithm provides a good approximation of the minimum evolution tree (the minimum-evolution method assumes the tree with the smallest sum of branch lengths is most likely to be the true one) [19]. Notably, the advantages of the neighbor-joining method are that it is fast (making it appropriate for use for larger datasets or bootstrap analysis), generates lineages with largely different branch lengths, allows for correction for multiple substitutions [20]. However, the main disadvantages of the algorithm are that different results may be obtained depending on the order in which sequences are inputted, generates only a single tree, and sequence information is reduced. Because branch lengths are presented as distances rather than discrete characters, it is impossible to identify important characters or infer ancestral states with this strategy.

Maximum Parsimony: Maximum parsimony methods use a matrix of discrete characters in order recover one or more optimal trees. These methods identify numerous equally parsimonious trees (minimizing number of steps and minimizes parallel nucleotides substitutions). Unlike neighbor-joining methods, maximum parsimony methods do not require a model of evolution and are able to identify important characters in the sequence and infer ancestral states throughout the reconstructed history. Maximum parsimony methods are relatively slower and run into issues when inputs contain highly unequal rates of base substitution, such as in protein domains when there are highly conserved sequences [20]. Maximum parsimony methods may not be statistically consistent, namely they do not converge on the correct answer with more data [20].

Maximum Likelihood: Maximum likelihood methods of tree construction use models of nucleotide or amino acid substitution like BLOSUM62 or JTT in order to reconstruct the tree topology as well as the branch lengths. Given a substitution model, maximum likelihood methods estimate the likelihood of that model being explained by the observed alignment of sequences. Maximum likelihood methods incorporate rate heterogeneity (the distribution of substitution rates amongst sites in the sequence), and thus use all of the data including invariable sites and unique mutations. These methods are the most flexible as input models can be parameterized with base frequencies and substitution rates, but are difficult to implement, and much slower than all other methods due to its computational intensity.

Bayesian Inference: Bayesian inference methods of phylogeny use a likelihood function called the posterior probability of a tree (the probability that the tree is correct) in order to produce the most likely phylogenetic tree. Bayesian inference methods evaluate the probability that a possible reconstruction is, given the data. Bayesian analyses typically assign probability values for observing specific data at specific sites, and returns a likely tree once a majority consensus tree is reached. Bayesian inferences are heavily dependent on the accuracy of these posterior probabilities, and unlike other methods utilize prior information in order to evaluate all possible combinations of substitution model parameter values and branch lengths [20].

2.3. Notable Algorithms

As previously described, there currently exist many widely available tools for phylogenetic tree reconstruction. Previous studies have indicated that even the most accurate gene tree reconstruction methods often produce erroneous estimates of the gene tree topology. Species tree construction benefits from large orthologous gene families, but accurate gene tree reconstruction is often confounded by limited genetic information, making it difficult for methods to reliably discern one gene tree topology over another [10] [8]. These problems are further exacerbated for protein domains, which are an even shorter subset of those sequences. There currently exist multiple tools which provide incomplete reconstruction methods, many of these tools do not take into account for any additional constraints introduced by protein domains.

TreeFix: TreeFix is a "integrative method" of tree reconstruction algorithm [9]. Integrative refers to utilizing a gene tree (which reflects the sequence level evolutionary history) as well as the species tree (which reflects the organism's evolutionary history) from which it was estimated. The "TreeFix" algorithm combines sequence data as well as species data in conjunction with maximum-likelihood methods in order to generate a "statistically equivalent gene tree that minimizes a species tree-based function" which is the most accurate tree topology [23]. Because TreeFix utilizes maximum likelihood equations, it has been shown to be computationally intensive and significantly slower than other hybrid methods as it lacks many optimizations available in other software packages [16]. TreeFix does not scale well for larger databases and often often assumes unrealistic loss and retention ratios.

RAxML: RAxML (Randomized Axelerated Maximum Likelihood) is a widely popular software program for phylogenetic analyses of large datasets. As the name suggests, RAxML is a maximum likelihood reconstruction method. RAxML offers four different ways to obtain bootstrap support to accelerate its search process. Notably, RAxML supports multiple data types (DNA, protein, RNA, multi-state morphological, and binary data), as well as a plethora of substitution models for each data type. In addition, RAxML offers a suite of software tools for post-analysis, including operations for calculating Robinson-Foulds distances, a variety of consensus trees (extended majority rule, majority

rule, and strict consensus trees) [21]. Despite these benefits, the accuracy of the reconstructed phylogenetic trees is constrained to the gene level. Given any arbitrary input, RAxML will utilize a standard substitution matrices for non-conserved regions in order to recreate the tree. For inputs containing protein domains or other relatively conserved region, however, this is not true, as specific amino acids in many functionally important protein domains are highly conserved. Thus, these reconstructions do not accurately reflect tandem duplications and other more advanced evolutionary behaviors exhibited by protein domains.

3. Approach

The main approach for reconstructing tree topologies was to identify different ways in order extract information about protein domains from the gene sequences in which they are found. The tree reconstruction algorithm is composed of a few major steps: 1) identifying protein domains in sequences 2) generating subtrees reflecting individual protein domain histories, and 3) combining individual protein domain histories in the highest probability ways. Notably, our approach centers around handling the proper histories for protein domains as distinct functional units in order to guide the topologies as opposed to conserved sequences which must be incorporated into pre-existing substitution models for genes. This attempts to ensure accurate reconstructions to handle the constraints associated with functional domain modeling.

In our previous research, we developed an accurate simulation for domain-level evolution inspired by reconciliation based approaches. Provided a host tree (representing gene-level events)m a guest tree (reflecting domain-level events), and a mapping of the guest tree to the host tree (to reflect duplications, losses, and horizontal gene transfers), the simulation was accurately able to replicate evolution of a domain through history; the sequence similarity of orthologous groups was consistent with actual sequence groups found in Eutheria [2]. We will evaluate the tree reconstructions against the simulated evolutionary history of protein domains, which will serve as a viable "ground truth" (since we do not know the true history) to gauge the accuracy of our reconstruction.

3.1. Identification, Alignment, and Initial Subtrees

The sequences at the leaves of the guest tree (which represent modern day domain-containing sequences) are aligned and identified using Clustal Omega helper functions (see Figure 3). This important first step separates the individual evolutionary histories of the protein domains. For input sequence groups only containing protein domains (therefore not including the non-coding linker regions surrounding the protein domains), we will use RAxML tree reconstruction methods in order to reconstruct multiple subtree topologies of protein domains are a variety of sites.

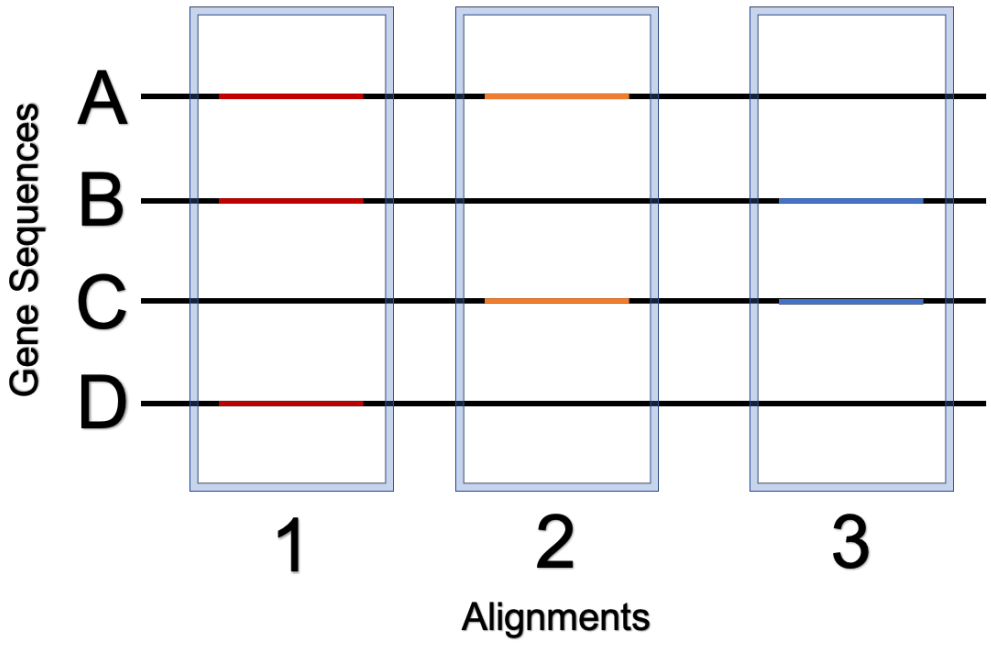

Figure 3: Identifying shared protein domains in guest tree leaves

3.2. Merging Individual Domain Histories during Reconstruction

As previously discussed, many protein domains emerged as a result of complicated multi-domain duplication and loss events. Domains exhibit behaviors including tandem duplication which closely tie pairs or groups of domains together. One way to resolve this behavior is by reconstruct the evolutionary history "in pairs". The similarity of the marginal ancestral sequence at the root of one domain subtree is compared against all of the marginal ancestral states of another domain subtree

to determine its closest potential sibling (see Figure 4). If necessary, an ancestral node is placed within the reconstruction to join the two trees (see Figure 5). This way, evolutionary histories of two different subtrees can be most accurately be combined at the point in which the evolutionary histories diverge, and is an intuitive explanation for the ties between associated protein domains.

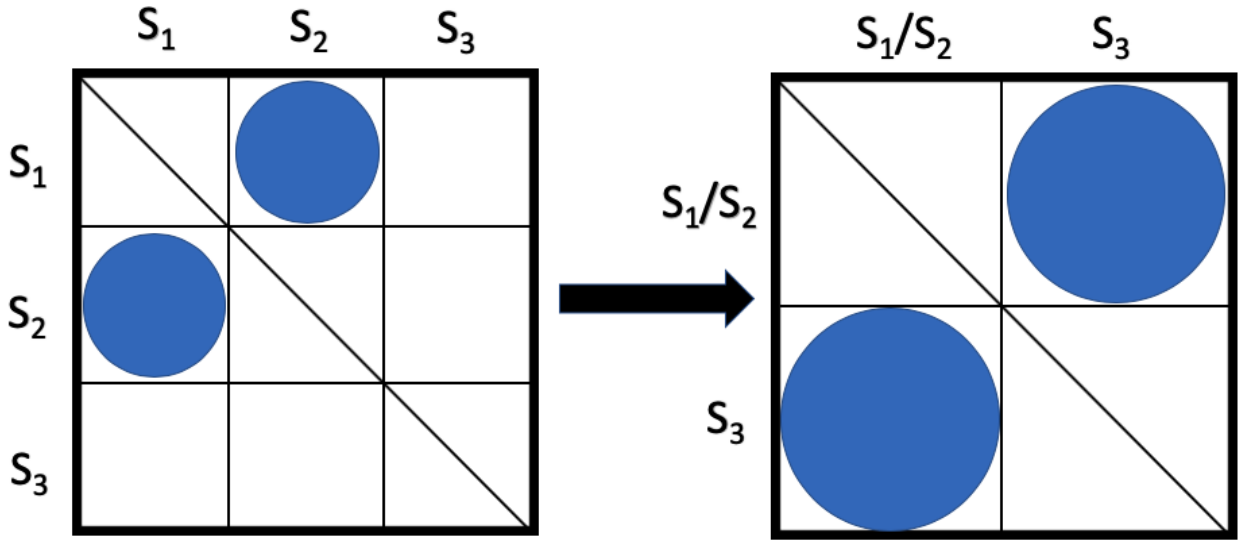

Figure 4: Determining order to combine individual domain histories

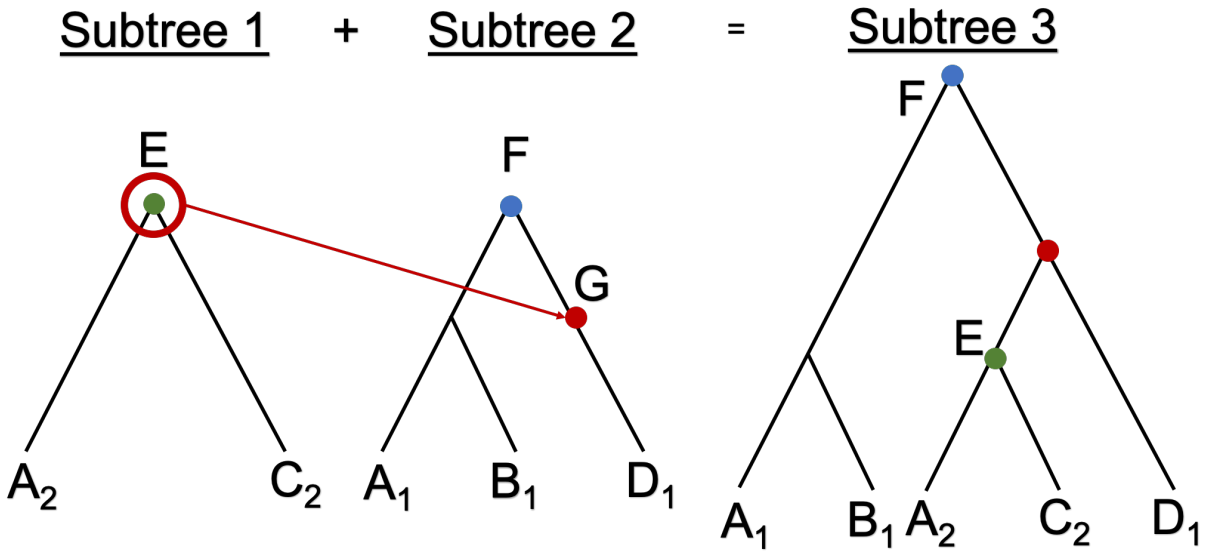

Figure 5: Combining domain histories based on sequence similarity

4. Implementation

Function Name and Parameters	Description
def findAndAlign(hmmfile, sequence)	Identifies and aligns protein domain sequences, and returns start positions, end positions, and sequences for protein domains
def generateRootedTree(sequenceFileName)	Performs midpoint rooting and generates a rooted tree from input sequence using RAxML
def generateMarginalAncestralStates(sequenceFileName)	Generates and returns output directory of marginal ancestral states of input sequence
def joinSubtree(maxTreeX, maxTreeY, maxNodeY, maxSimilarity, fastaFiles)	Recursively returns a list of fasta files of unjoined subtrees during tree reconstruction
def branchLengths(sequenceFileName)	Returns branch annotated Newick tree file for an input sequence

Table 1: Reconstruction API

Table 1 details the API that was developed for the tree reconstruction algorithm. As described in the approach, the implementation for this tree reconstruction algorithm was divided into finding and aligning protein domains, generating rooted subtrees using RAxML, and performing a variety of calculations to connect and join those subtrees.

4.1. Birth-Death Process

The main implementation of this algorithm was developed in Python 2.7. Python features robust libraries for managing tree based data structures (ETE3), as well as easy-to use libraries for common computational biology tasks such as identifying protein domains, sequence alignments (including support for Clustal Omega and HMMER profiels) and other evolution processes. ETE3 is a Python framework which provides a rich API for the annotation, analysis, and visualization of trees. ETE3 allows us to create, search, and modify our guest trees and host trees as Python objects. ETE3 was the main algorithm utilized by the previous research to generate domain evolution simulations which serve as points of comparison for the reconstruction algorithm. Figure 6 is a screenshot taken

of the host and guest trees and various associated node features, including node label, branch length, and node events, which were created using ETE3.

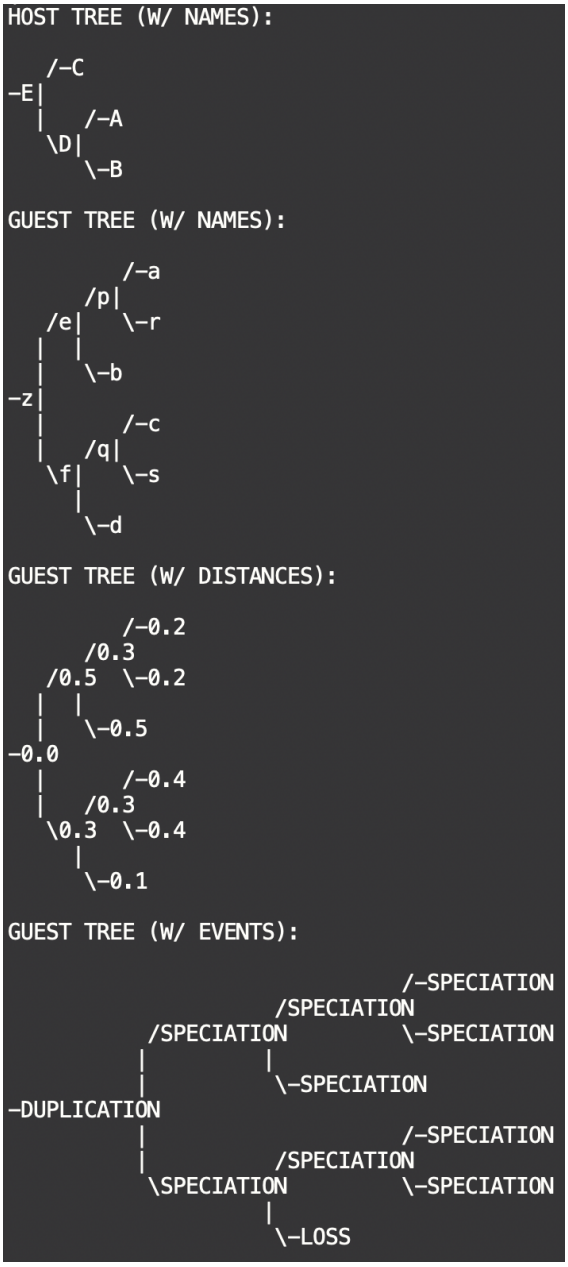

Figure 6: Example of a ETE3 structure host tree

The host trees and guest trees used for parts of the analysis were developed by advisor Chaitanya Aluru according to the birth-death models of diversification [**?**]. Birth-death models are defined as a continuous-time Markov process which tracks how individuals change over time based on a given birth and death rate. Despite the fact that any random binary topologies could be used to generate

a guest tree to evaluate our phylogenetic tree reconstruction, by using the birth-death paradigm, multidomain duplications can better be simulated and reflected in sequence groups. Birth-death models are particularly useful when performing phylogenetic analysis; birth-death models intuitively represent events where births reflect duplications (number of "individuals" or species increases by one) and deaths reflect losses (number of "individuals" or species decreases by one) [1].

The current software suite enabled by the previous semester's research includes a host tree generation function which takes in birth rate, death rate, and tree height to randomly create a tree topology [2]. The software assigns branch lengths all of the nodes of the tree. From this host tree, an additional function in the simulation software suite creates a guest tree topology. This function creates a guest tree within each node of the host tree based on the input host topology, duplication rates, and average distances between events. Along with a mapping mapping between the nodes of the tree, this guest tree is used as a benchmark for subsequent analysis and evaluation of the phylogenetic tree reconstructions.

For the purposes of serving as evaluative benchmark for this experiment, multiple iterations of host tree and guest trees are generated. The simulation outputs an orthogroup as well as bookkeeping file which tracks the start positions, end positions, and sequences of all protein domains involved which can be used to rederive the associated guest tree leaves (notably containing only the sequences of protein domains), which serve as the the inputs for the tree reconstruction algorithm. Sequence alignment and protein domain identification is carried out using Clutal Omega and other software packages previously devloped in the simulation suite. These sequences are outputed along with the identical names of the guest tree nodes to a FASTA file for ease of readability in the remainder of the algorithm.

4.2. Midpoint Rooting

Because RAxML typically generated unrooted trees, we must perform midpoint rooting in order to generate a rooted tree for our subsequent marginal ancestral state calculations. One way of doing this is by modifying the original FASTA file and append a fictional outgroup to the FASTA file. The

input file is converted to and from a dictionary, and an entry titled OUTGROUP whose sequence is completely randomly generated is added. For each input FASTA file, a rooting "OUTGROUP_" file is generated. RAxML is called utilizing the PROTGAMMA amino acid substitution matrix and the BLOSUM62 substitution model. The PROTGAMMA parameter specifies an amino acid substitution matrix and the GAMMA model of rate heterogeneity. A unrooted tree is generated, and is rooted by pruning the fictitious outgroup. This rooted tree reflects the evolutionary history of a single protein domain.

4.3. Generating Marginal Ancestral States

Marginal reconstruction of ancestral states refers to finding the sequence at a node that maximizes the likelihood of integrating over all nodes. Calculating marginal ancestral states is particularly useful for a variety of analyses by providing evidence about the way sequences change throughout their evolutionary history. RAxML has a flag (-t) for computing marginal ancestral states on a rooted tree, when provided a FASTA file and a ROOTED reference tree. The output of the midpoint rooting is passed into RAxML once again, and node labelled rooted trees (including internal nodes), as well as marginal ancestral states for those nodes. These marginal ancestral states are directly utilized in order to connect individual protein domain histories.

4.4. Connecting Subtrees

Now that we have reconstructed individual histories for each of the individual protein domains, we will now combine these trees together in the most optimal way, taking into consider the sequence similarities of the marginal ancestral states. For each pair of subtrees, the ancestral sequence at the root of one subtree is compared to every internal node of another subtree via a domain similarity function developed in the simulation suite (see Figure 4). For the highest similarity pair, the root of one subtree are joined in a way such that the root becomes a sibling of its counterpart in the other tree (see Figure 5). Since this tree is also rooted, we can recursively recalculate the marginal ancestral states for this subtree and compare it to any other remaining trees until we are left with one full tree reconstruction. Since we directly utilized the protein domain sequences of at the leaves

of the guest trees from a protein domain simulation, we have identical leaf sequences and can subsequently compare the generated trees with original trees in order to determine the accuracy of the tree reconstruction algorithm.

5. Evaluation and Results

500 pairs of randomized corresponding host and guest trees were generated using the domain simulation. Host trees were generated using the birth-death process with input parameters of a duplication rate of 0.3, a loss rate of 0.1, and an average tree height of 0.5, and trees guaranteed to have at least 25 nodes. Observations from the simulations last year indicated that due to random chance, degenerate tree structures with single node trees could be generated at low frequencies from the simulation [2]; because RAxML can only reconstruct trees with at least 8 input species, ensuring valid trees was incorporated as a safety measure. Guest trees were generated using a sigmoid function for duplication rate, and an exponential function for duplication size. The specific simulation chosen as an input for the tree reconstruction utilized the standard C2H2 zinc finger HMM model and the same randomly generated root sequence for each of these 500 pairs of trees.

One way of evaluating the similarity of these trees is the Robinson-Foulds distance. The Robinson-Foulds distance is defined as the number of bipartitions that exist between two trees; for a pair of trees it can be calculated by summing the number of partitions implied by both trees but not the other. The Robinson-Foulds distance often serves as a tool in phylogeny in order to evaluate differences between trees. ETE3 has a built in package for calculating Robinson-Foulds distances between trees which was used for the comparisons in this experiment.

We calculated distances between the simulated trees and the reconstructed trees across all pairs of these simulations, and plotted the results. Figure 7 and Figure 8 demonstrate the distributions of standard Robinson-Foulds distances and normalized Robinson-Foulds distances between the simulated trees and the reconstructed trees. The graph of the standard Robinson-Foulds distances was a relatively bell-shaped distribution. The distribution is negatively skewed, indicating that the median is greater than the mean for this distribution. The mean Robinson-Foulds distance was 62.61.

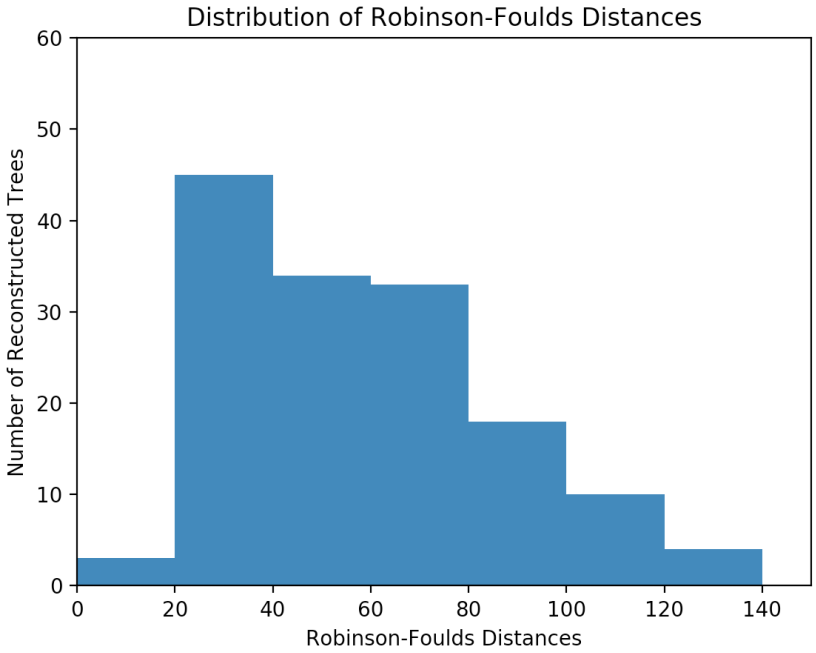

Figure 7: Simulated Distribution of Robinson-Foulds distances

A particularly interesting statistic from these inferences is a very large standard deviation (even over five-hundred simulation reconstruction processes); we observed a standard deviation of 243.78. This large variance suggests inconsistencies depending on sequence group input for topology of

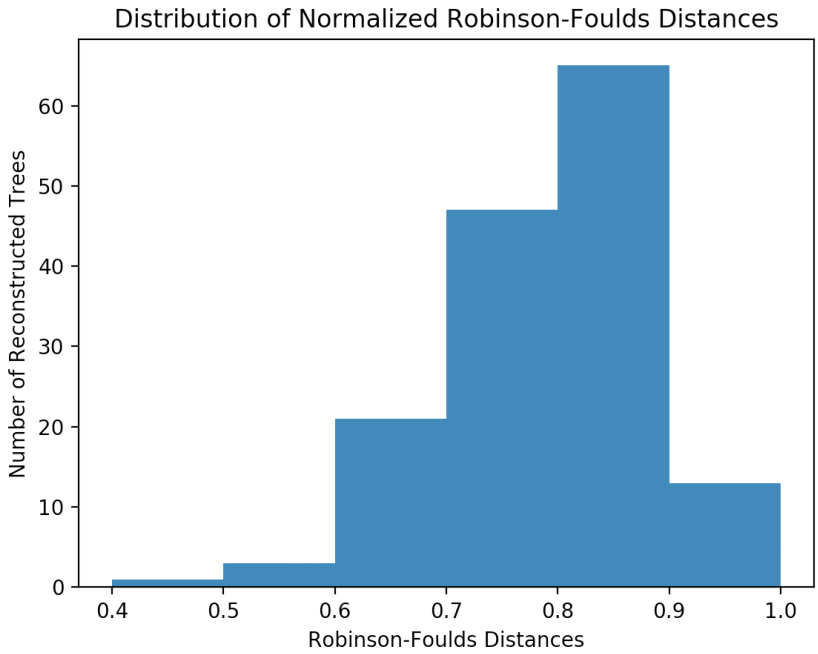

Figure 8: Simulated Distribution of Normalized Robinson-Foulds distances

tree generated by the algorithm. Normalized Robinson-Foulds distances (on a scale from 0 to 1) indicate similar levels of accuracy with reconstructing evolutionary trees, with an average distance of 0.789 with variance 0.009.

The standard Robinson-Foulds distances relative to the size of the tree (host tree initialized with at least 25 nodes) and the high normalized Robinson-Foulds distance (0.789) leads us to believe that this reconstruction method may not accurately capture the evolutionary history of conserved domains. One of the possible main causes of these differences in simulation and reconstruction are because RAxML uses a different model of sequence evolution than the previous simulations. The domain simulations are based on HMM files. HMM files reveal the conservation pattern for a set of sequences and indicate the probabilities of observing all possible amino acids at those positions. By using RAxML to build parts of the overall domain history, we have constrained reconstruction to not properly handle the problems associated with domain-level events. Given these findings, we evaluated the performance of our reconstruction algorithm against that of standard RAxML.

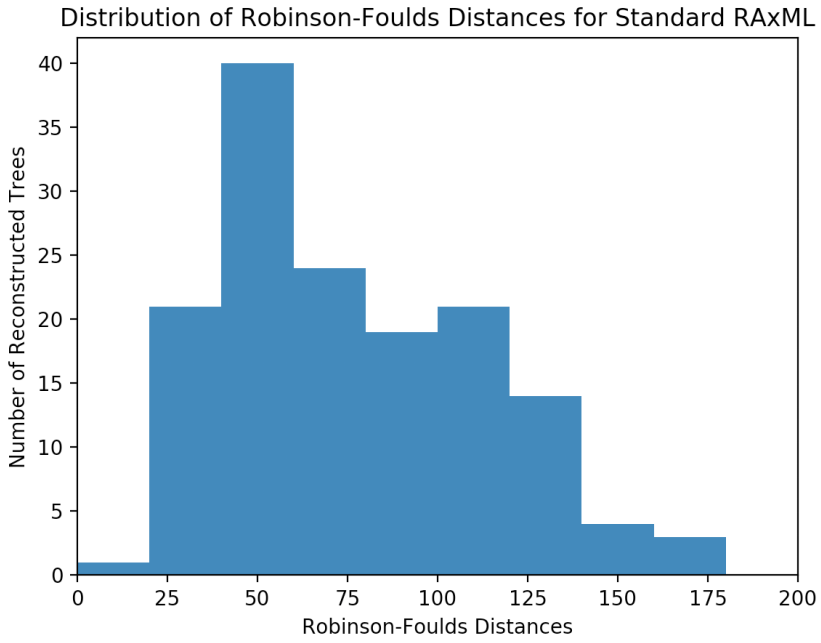

Figure 9: Simulated Distribution of Robinson-Foulds distances as generated by standard RAxML

We ran conventional RAxML on the full sequences (not just the domains) to compare its performance against our algorithm. Comparing the graphs in Figure 9 and Figure 10 to the previous

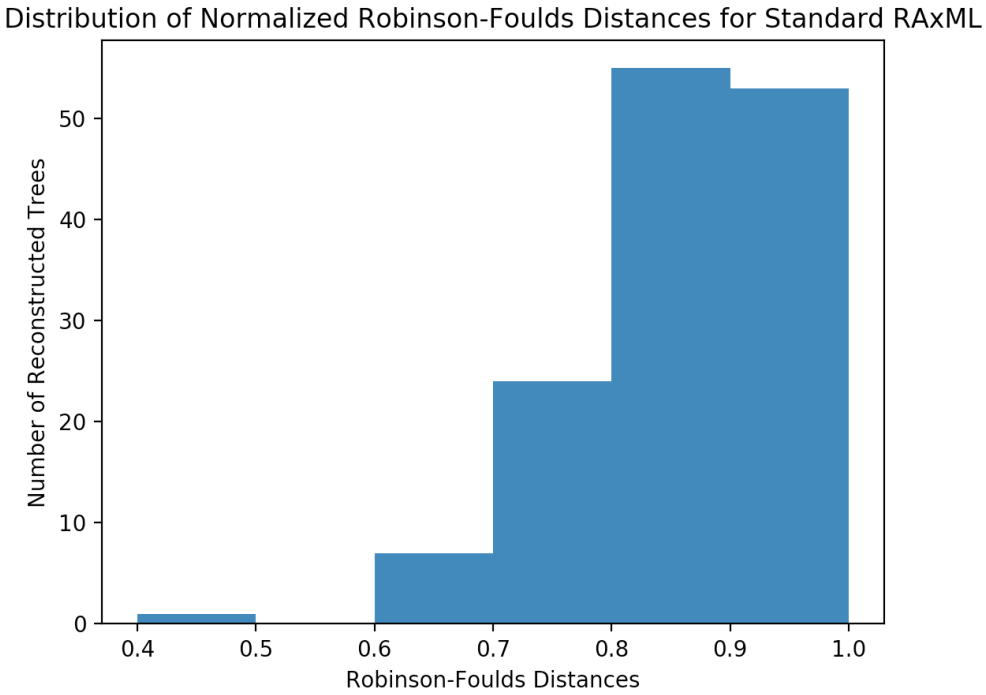

Figure 10: Simulated Distribution of Normalized Robinson-Foulds as generated by standard RAxML

graphs, we notice that our algorithm slightly outperforms the conventional RAxML at reconstructing protein domain topologies for this specific simulation. The averages of standard and normalized Robinson-Foulds distances were 81.365 and 0.867 respectively, thus indicating there for protein domain simulations generated by the zinc finger HMM file, our reconstruction actually performs better than widely available options including RAxML. Though our model of joining subtrees if fairly simplistic, it may better capture the intricacies of domain duplications and other behavior when compared to typical reconstruction algorithms.

6. Conclusion

The tree reconstruction algorithm developed in this independent work provides contributes to the suite of different phylogeny reconstruction algorithms used by many researchers today. Previous research in developing protein domain simulations laid the groundwork to design a new algorithm for reconstructing tree topologies. The data from Robinson-Foulds distance calculations indicates that despite struggling to generate accurate trees for the specific simulation [2], the new algorithm

outperforms traditional RAxML algorithms at the same task. By developing a reconstruction process which takes into account domain behavior and constraints, we have gained insight on types of methods which can improve reconstruction accuracy. It is particularly important to consider the context of this experiment, namely that these observations were made for a single specific simulation model. Future goals of this research are to expand testing of this reconstruction algorithm on multiple other types of protein domains (even those that exhibit much more sophisticated behavior including lateral domain transfers and domain splits) and other methods of designing simulation models.

7. Acknowledgements

I would like to thank my advisors Mona Singh and Chaitanya Aluru for their continued support of my interests in computational biology this semester and their constant advice and support during my independent work. Thank you for giving me an opportunity to learn amazing new things and to challenge myself with interesting and important research questions. This is my last semester at Princeton, and I thoroughly enjoyed what I love love the most, at the place I love the most.

I pledge my honor that this paper represents my own work in accordance with University regulations.

References

[1] "Birth Death Process," https://lukejharmon.github.io/pcm/chapter10_birthdeath/#section-10.2-the-birth-death-model4, accessed: 2019-04-12.

[2] "Simulating domain-level evolution in zinc fingers," 2019.

[3] M. A. Andrade, C. Perez-Iratxeta, and C. P. Ponting, "Protein repeats: structures, functions, and evolution," *Journal of structural biology*, vol. 134, no. 2-3, pp. 117–131, 2001.

[4] G. Apic, J. Gough, and S. A. Teichmann, "Domain combinations in archaeal, eubacterial and eukaryotic proteomes," *Journal of molecular biology*, vol. 310, no. 2, pp. 311–325, 2001.

[5] J. G. Apic, Gordana and S. A. Teichmann., "An insight into domain combinations." *Bioinformatics*, vol. 17, no. 1, pp. S83–S89, 2001.

[6] e. a. Becker, Annette, "Mads-box gene diversity in seed plants 300 million years ago." *Molecular Biology and Evolution*, vol. 17, no. 10, pp. 1425–1434, 2000.

[7] Å. K. Björklund, D. Ekman, and A. Elofsson, "Expansion of protein domain repeats," *PLoS computational biology*, vol. 2, no. 8, p. e114, 2006.

[8] e. a. Burleigh, J. Gordon, "Genome-scale phylogenetics: inferring the plant tree of life from 18,896 gene trees." *Nature Reviews Genetics*, vol. 60, no. 2, pp. 117–125, 2011.

[9] e. a. Christensen, Sarah, "Mtraction: Fast non-parametric improvement of estimated gene trees." *19th International Workshop on Algorithms in Bioinformatics*, 2019.

[10] H. B. Delsuc, Frederic and H. Philippe., "Phylogenomics and the reconstruction of the tree of life." *Nature Reviews Genetics*, vol. 6, no. 5, pp. 361–375, 2005.

[11] e. a. Demuth, Jeffery P., "The evolution of mammalian gene families." *PloS one*, vol. 1, no. 1, 2006.

[12] P. Djian, "Evolution of simple repeats in dna and their relation to human disease," *Cell*, vol. 94, no. 2, pp. 155–160, 1998.

[13] R. O. Emerson and J. H. Thomas, "Adaptive evolution in zinc finger transcription factors," *PLoS genetics*, vol. 5, no. 1, p. e1000325, 2009.

[14] C. Looman *et al.*, "Krab zinc finger proteins: an analysis of the molecular mechanisms governing their increase in numbers and complexity during evolution," *Molecular biology and evolution*, vol. 19, no. 12, pp. 2118–2130, 2002.

[15] E. M. Marcotte *et al.*, "A census of protein repeats," *Journal of molecular biology*, vol. 293, no. 1, pp. 151–160, 1999.

[16] e. a. Noutahi, Emmanuel, "Efficient gene tree correction guided by genome evolution." *PLoS One*, vol. 11, no. 8, 2016.

[17] L. R. Pal and C. Guda., ""tracing the origin of functional and conserved domains in the human proteome: implications for protein evolution at the modular level." *BMC evolutionary biology*, no. 6.1, p. 91, 2006.

[18] C. P. Ponting and R. R. Russell., "The natural history of protein domains." *Annual review of biophysics and biomolecular structure*, vol. 31, no. 1, pp. 45–71, 2002.

[19] A. Rzhetsky and M. Nei., "Theoretical foundation of the minimum-evolution method of phylogenetic inference." *Molecular Biology and Evolution*, vol. 10, no. 5, pp. 1073–1095, 1993.

[20] D. E. Soltis and P. S. Soltis., "The role of phylogenetics in comparative genetics." *Plant Physiology*, vol. 132, no. 4, pp. 1790–1800, 2003.

[21] A. Stamatakis, "Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies," *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.

[22] M. Toll-Riera and M. M. Alba., "Emergence of novel domains in proteins." *BMC evolutionary biology*, vol. 13, no. 47, 2013.

[23] M. S. B. M. K. Yi-Chieh Wu, Matthew D. Rasmussen, "Treefix: Statistically informed gene tree error correction using species trees," *Systematic Biology*, vol. 62, no. 1, 2013.