Research Article

AminoGraph Analysis of the Auditory Protein Prestin From Bats and Whales Reveals a Dependency-Graph Signal That Is Missed by the Standard Convergence Model

Winston Ewert*
Biologic Institute, Redmond, Washington, USA

Abstract

Alternative models to the theory of universal common descent have, thus far, been underdeveloped. Our previous work introduced a *dependency graph* model as an alternative way of explaining the patterns of genetic similarity and diversity among living things. According to this model, different forms of life share similarities because they share function-specific genetic features (modules) that may have dependencies on other genetic features. Here, we introduce a tool (*AminoGraph*) that infers dependency graphs from protein sequence alignments, and we apply this to prestin, a mammalian auditory protein that requires special modifications for ultrasonic hearing in species that use echolocation. Prestin sequences from some echolocating bats show similarities with prestin sequences from echolocating whales. Conventional analyses interpret this as convergence, not because convergence is known to be evolutionarily feasible, but because this preserves the presumed phylogenetic tree. The AminoGraph analysis of prestin presented here provides an alternative explanation: echolocation is supported by two prestin-modifying modules, one or both of which are seen in all echolocating bats and whales. The reliability of this inference is increased by thorough testing of AminoGraph on generated test data sets where sequences are either unrelated, related by common descent, or related by deployment of modules. In all cases, AminoGraph produces the expected relationships.

Cite As: Ewert W (2023) AminoGraph Analysis of the Auditory Protein Prestin From Bats and Whales Reveals a Dependency-Graph Signal That Is Missed by the Standard Convergence Model. BIO-Complexity 2023 (1):1-15. **doi:**10.5048/BIO-C.2023.1.

Editor: Robert J. Marks II

Received: January 30, 2023; **Accepted:** May 8, 2023; **Published:** August 8, 2023

Notes: A *Critique* of this paper, when available, will be assigned **doi:**10.5048/BIO-C.2023.1.c.

*Email: evoinfo@winstonewert.com

1. INTRODUCTION

In our previous work [1], we introduced the dependency graph of life as an alternative model to universal common descent. According to this model, different forms of life share similarities due to sharing common modules much the same way different software programs share similarities and code due to reusing common modules. The modules used in a particular organism or program are constrained by a dependency graph. In a dependency graph, some modules depend on other modules, such that a module cannot be included without also including its dependencies. Under the theory of universal common descent, taxonomic categories are defined by the most recent common ancestor of all species within that category. In contrast, under the dependency graph model, taxonomic categories are defined by a module on which all species in that category depend either directly or transitively.

Our previous work [1] evaluated the model by using Bayesian model selection to determine whether the distribution of gene families across species better fit a tree, as predicted by common descent, or a dependency graph. It was determined that across nine different gene databases—each using their own system for classifying genes into gene families—the data fit the dependency graph model overwhelmingly better than that of a tree. However, this was only a first step in evaluating whether or not this model could explain the pattern of similarities and differences found in living things.

This paper expands on previous results, transitioning from binary datasets (presence-absence) of particular gene families in particular genomes to datasets of amino acid alignments. The accuracy of presence-absence data is limited both by the accuracy in identifying all of the genes found in a particular species' genome and by the accuracy in classifying those genes correctly into gene families. It is possible that the apparent success of the dependency graph model on those datasets was an artifact of inaccurate gene-identification and classification data in the databases. Nevertheless, that outcome would require those possible inaccuracies to induce a dependency graph pattern in the resulting data, and it is not clear how or why that should be the case.

More crucially, binary data sets are not the sort of dataset from which common descent is usually inferred. Papers seeking to test the hypothesis of common ancestry typically utilize genetic sequences [2–5]. They do not consider whether or not two species share a similar gene, but consider the pattern of similarities and differences in the sequence of amino acids in a protein or nucleotides making up a gene. The dependency graph model must account for these patterns if it is to be a successful theory.

Further improvements include more accessible results and higher quality inferred graphs. The previous work provided a detailed description of the algorithm used to infer the dependency graph, but relatively few would have the ability to implement the algorithm themselves. This work provides a new tool: *AminoGraph*. This tool allows interested researchers to analyze any amino acid alignment. The inferred graphs for the original paper were also provided as supplemental files, but these would be challenging for most readers to interpret. Furthermore, the broad scope of the data evaluated along with the resulting complexity of the inferred graphs made it challenging to draw useful conclusions from the results. Instead, this work focuses on a small example, that of prestin in echolocating species, and thus, provides an easier to grasp example and illustration of the model. The previous work was the first attempt to infer a dependency graph. The purpose was to demonstrate that a graph could be fit to the data rather than attempting to infer the best possible graph. As such, drawing conclusions from the inferred graphs would be dubious. However, with *AminoGraph*, attention has been paid to attempt to develop the model to produce useful and illuminating results.

First, in Section 2, we will motivate the necessity of a model like the dependency graph to account for the problem of discordant phylogenies where different proteins—and different amino acids within proteins—give conflicting signals when attempting to infer a phylogenetic tree. We will specifically consider how this situation plays out for the prestin protein in echolocating and non-echolocating whales and bats. In Section 3, we will explain the dependency graph model for amino acid sequences. In Section 4, we will look at results obtained by analyzing amino acid alignments using the *AminoGraph* tool. We will look at how it performs on random, simulated, generated, and biological data. Section 6 will discuss the conclusions that can be drawn from the results. Section 7 is an appendix which will provide the details of the probabilistic model.

2. THE PROBLEM: DISCORDANT PHYLOGENIES

Penny et al. (1982), in an early attempt to statistically verify common descent using genetic sequence data, wrote [3]:

> The theory of evolution predicts that similar phylogenetic trees should be obtained from different sets of character data.

More recent papers do not make this prediction. They instead simply state that phylogenetic trees inferred from different genes or proteins are often in conflict [6–10].

The most frequently given explanation for this conflict is incongruence between the species tree and the gene tree [11]. This divergence arises for a number of different reasons including incomplete lineage sorting, horizontal gene transfer and gene duplication/extinction. If every gene were a simple, possibly mutated, copy of the same gene in the organism's immediate ancestor, then the gene tree and species tree would be the same. However, because that gene might derive from either the mother or father of the organism, some other organism entirely or another copy of the gene, the gene tree is not necessarily identical to that of the organism as a whole.

The secondary explanation cited is convergent evolution, in which the same changes are gained or lost in multiple lineages, clouding the phylogenetic signal. This could happen simply by random chance. Given sufficient data, there will certainly be cases in which, purely by coincidence, the same mutation occurs multiple times. Alternatively, if selection favors certain mutations, this would help explain why the same mutation would be preserved in multiple lineages.

To better illustrate the issue, we will consider the case of echolocating mammals (bats and cetaceans), whose relationships are depicted in Figure 1. Traditionally, bats were divided into two groups: megabats and microbats. Megabats are larger, eat fruit, and do not use echolocation: examples include the fruit bats and flying foxes. Microbats are smaller and use echolocation: examples include horseshoe bats and the vesper bats. However, on the basis of genetic analysis, bat phylogeny has been reorganized. The new clade Pteropodiformes is divided into Pteropodidae, containing all the megabats, and Rhinolophoidea, which contains some of the microbats, including the horseshoe bats. Vespertilioniformes is a sister clade to Pteropodiformes and contains the remaining microbats, including the vesper bats.

For the cetaceans, the situation is simpler. They are divided into the toothed whales—Odontoceti—which engage in echolocation and the baleen whales—Mysticeti—which do not.

There are thus three clades which utilize echolocation: Vespertilioniformes, Rhinolophoidea and Odontoceti. According to the standard evolutionary account, all three groups evolved echolocation separately. Bats and cetaceans are distant relatives under the theory of common descent. Bats are held to have diverged from the line leading to cetaceans early in mammalian history. Numerous other clades, leading to many well-known species of mammals, have branched away from the line leading to cetaceans since the divergence from bats.

Prestin is a motor protein used in hair cells in the inner ear. Previous work by another group reported that a phylogeny inferred from prestin's amino acid sequence brought all three echolocating groups together—both clades of microbats and the toothed whales [12, 13]. In order to confirm this result, we obtained a collection of prestin sequences by combining sequences listed in Ensemble [14], OrthoDB [15], and those referenced in previous papers that discussed their similarities [12, 13]. We used the program Clustal W [16] to align these sequences and obtain a phylogenetic tree using the nearest-neighbor-joining algorithm. Figure 2 depicts the portion of the tree containing the bats and cetaceans. The cetaceans have been placed in the middle of the microbats in the inferred phylogenetic tree. The bats in Rhinolophoidea have been grouped with the other microbats in Vespertilioniformes, instead of with the megabats in their sister clade Pteropodidae. In accordance with the previous research [12, 13], the echolocating species were again grouped together in the phylogenetic tree inferred from prestin.

A simple way to measure the fit of sequence data to a tree is to count the minimum number of substitutions, insertions and deletions required to obtain the observed sequences. Maximum-parsimony methods of phylogenetic inference aim to select the tree with the smallest number of such changes. We compared three trees constructed by placing the Rhinolophoidea clade into three different positions: 1) the standard tree where Rhinolophoidea is a sister clade to Pteropodidae, the megabats, 2) a tree where Rhinolophoidea is placed with the other microbats, and 3) a final tree where Rhinolophoidea is placed as a sister clade to Odontoceti, the echolocating toothed whales. Instead of attempting to determine the single tree which best fits the alignment, we determined how many changes each individual position of the alignment required. The results are shown in Table 1.

A number of loci are a better fit to a phylogeny which brings together the microbats than in the standard phylogeny, as seen in columns 1, 2, 7, and 12. Inspection of the amino acids for these columns shows that the microbats tend to be similar to each other. However, column 19 suggests the opposite, that Rhinolophoidea and Pteropodidae are more similar. Another set of loci are a better fit to a phylogeny which places Rhinolophoidea alongside Odontoceti, as seen in columns: 3, 4, 5, 8, 10, 13, 16, 17, and 18. Looking at the amino acids, we see that the sequence for Rhinolophoidea has many similarities to Odontoceti. However, many of those loci are not restricted to Odontoceti but are found in the baleen whales and even Suina (pigs). Nevertheless, there are loci which present the opposite signal where Rhinolophoidea, as might be expected, is more similar to the other bats. This can be seen in loci 6, 9, 11, 14, and 15, where the three groups of bats tend to be similar.

There are three strong signals that can be seen in this prestin sequence. Firstly, the microbats, Vespertilioniformes and Rhinolophoidea show similarities in their amino acid sequences. Secondly, the Rhinolophoidea and Odontoceti have similarities in their amino acid sequences. Thirdly, all of the bats have similarities in their amino acid sequences. On the other hand, there is very weak signal showing similarities among the Pteropodiformes, that is Rhinolophoidea and Pteropodidae.

There is only a weak signal bringing all of the echolocating species together. There are not many similarities which are common to all echolocating species. This is somewhat surprising because we saw in Figure 2 that all of the echolocating species are grouped together. This, however, does not indicate a common signal between all echolocating species. Rather, it reflects two signals, one linking Odontoceti and Rhinolophoidea, and the other linking Rhinolophoidea and Vespertilioniformes. The only way for a tree to reflect both signals is to bring all three groups together.

Table 2 summarizes these results and the fit of these trees at the nucleotide level. The signals previously discussed are present at both the amino acid and the nucleotide levels. However, the signal favoring placing Rhinolophoidea within the bats and specifically with Pteropodidae is much stronger at the level of nucleotides. Whereas the strongest signal at the amino acid level favors grouping at least Rhinolophoidea with the echolocating toothed whales (Odontoceti) the strongest signal at the nucleotide level favors placing them with the other bats. Indeed, phylogenetic inference based on nucleotides for this alignment much more closely agrees with evolutionarily expected phylogeny. Typically, phylogenetic inference would follow the strongest signal, but this does not tell the whole story. The other signal, while weaker, still exists. To account for the data, any theory must account for both signals.

The conclusion to be drawn is that the problem of discordant phylogenies is not simply that some genes or proteins suggest different phylogenetic trees than the generally accepted species tree. Rather, we find that

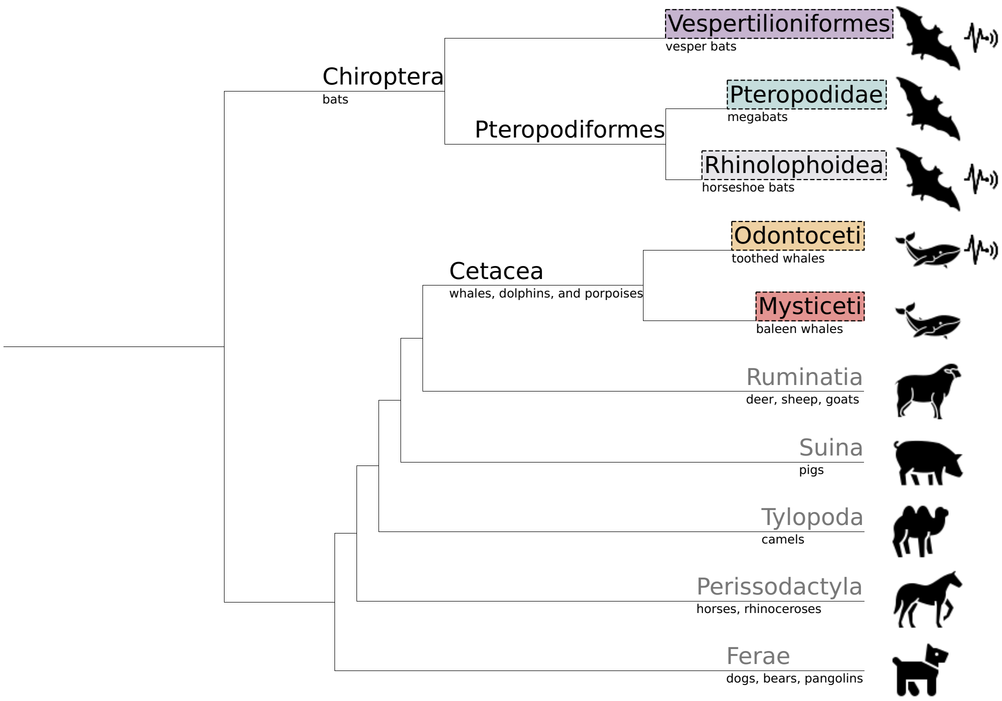

Figure 1: Phylogenetic tree showing the currently understood relationship between the echolocating clades and closely related non-echolocating clades. The five clades of interest of bats and cetaceans are each given a unique color which will be used throughout the paper. The icon next to Vespertilioniformes, Rhinolophoidea and Odontoceti denotes that these clades use echolocation. doi: 10.5048/BIO-C.2023.1.f1

Table 1: Alignment of loci in which relocating the Rhinolophoidea clade changes the minimum required number of amino acid changes to account for the observed alignment. The icons and colors used for each clade correspond to Figure 1 for easier comparison. The three rows on the bottom give the minimum number of amino acid changes required to account for the data in the three different trees. The numbers corresponding to the tree with the smallest number of changes are marked in bold.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Myotis lucifugus	T	L	P	I	A	M	S	L	A	D	D	I	I	I	V	N	S	K	P
Myotis brandtii	T	L	P	I	A	M	S	L	A	D	D	I	I	I	V	N	S	K	P
Myotis ricketti	T	L	P	I	A	M	S	F	A	D	D	I	I	I	V	N	S	K	P
Eptesicus fuscus	T	L	P	I	S	M	S	L	A	D	D	V	I	I	V	N	S	K	P
Miniopterus fuliginosus	T	L	P	I	S	M	S	L	A	D	E	I	I	F	V	N	Q	K	P
Miniopterus natalensis	T	L	P	I	S	M	S	L	A	D	E	I	I	F	V	N	Q	K	P
Megaderma spasma	T	S	S	I	S	M	S	L	A	D	D	I	T	F	V	N	Q	L	P
Rhinolophus pusillus	T	L	S	T	A	M	S	F	A	E	D	V	T	F	V	S	R	L	T
Rhinolophus sinicus	T	L	S	T	A	M	S	F	A	E	D	V	T	F	V	S	R	L	T
Rhinolophus ferrumequinum	T	L	S	T	A	M	N	F	A	E	D	V	I	F	I	S	R	L	T
Rhinolophus luctus	T	L	S	T	A	M	S	F	A	G	D	V	I	F	V	S	Q	L	M
Hipposideros armiger	T	L	S	T	A	M	N	F	A	E	E	I	I	F	V	S	R	L	T
Hipposideros larvatus	T	L	S	T	A	M	N	F	A	E	E	I	I	F	V	S	R	L	T
Hipposideros pratti	T	L	S	T	A	M	N	F	V	E	E	I	I	F	V	S	R	L	T
Aselliscus stoliczkanus	T	L	S	T	A	M	S	F	A	E	E	I	T	L	V	S	R	L	T
Delphinapterus leucas	S	P	S	T	A	L	S	F	V	E	N	I	T	L	I	S	R	L	P
Monodon monoceros	S	P	S	T	A	L	S	F	V	E	N	I	T	L	I	S	R	L	P
Orcinus orca	T	P	S	T	A	L	S	F	V	E	D	I	T	L	I	S	Q	L	P
Tursiops truncatus	T	P	S	T	A	L	S	F	V	E	D	I	T	L	I	S	Q	L	P
Phocoena sinus	S	P	S	T	A	L	C	F	V	E	D	I	T	L	I	S	R	L	P
Lipotes vexillifer	S	P	S	T	A	L	S	F	V	E	N	I	T	L	I	S	R	L	P
Physeter catodon	N	P	S	T	A	L	S	F	V	E	N	I	T	L	V	D	R	L	P
Balaenoptera musculus	N	P	S	I	S	L	S	L	V	E	N	I	I	L	I	N	R	L	P
Balaenoptera acutorostrata scammoni	N	L	S	I	S	L	S	L	V	E	N	I	I	L	I	N	R	L	P
Rousettus leschenaultii	N	P	P	I	S	M	N	F	A	-	D	V	I	F	V	N	Q	R	T
Rousettus aegyptiacus	N	P	P	I	S	M	N	F	A	-	D	V	I	F	V	N	Q	R	T
Pteropus vampyrus	N	P	P	I	S	M	N	L	A	-	D	V	I	F	V	N	Q	R	A
Pteropus alecto	N	P	P	I	S	M	N	L	A	-	D	V	I	F	V	N	Q	R	A
Cynopterus sphinx	N	P	P	I	S	M	N	L	A	-	D	V	I	F	V	N	Q	R	T
Sus scrofa	N	P	S	I	S	L	N	L	V	E	D	I	I	F	I	D	R	R	P
Catagonus wagneri	N	P	S	I	S	L	N	L	V	V	D	I	I	F	I	D	R	R	P
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Standard Tree	10	6	11	5	5	**1**	14	23	**16**	20	**18**	17	13	**12**	**19**	8	14	11	**10**
Rhinolophoidea with microbats	**9**	**5**	11	5	5	**1**	**13**	23	**16**	20	**18**	**16**	13	**12**	**19**	8	14	11	11
Rhinolophoidea with Odontoceti	10	6	**10**	**4**	**4**	2	**13**	**22**	17	**19**	19	**16**	**12**	13	20	**7**	**13**	**10**	11

Table 2: The number of nucleotide and amino acid loci for which each tree has the best and worst minimum number of required nucleotides or amino acid changes. This summarizes how often each tree does a better or worse job of accounting for the observed amino acid and nucleotide sequences than the two alternatives. Generally, there are approximately twice as many nucleotide loci as amino acid loci for which a given tree requires either fewer or more changes than the alternatives. The biggest exception is placing Rhinolophoidea with Odontoceti. This requires additional changes for ten times as many nucleotides as amino acids. A smaller exception is that moving away from the standard tree requires six nucleotide substitutions but only one amino acid substitution. The signal favoring placing Rhinolophoidea with the other bats, and Pteropodidae specifically, is much stronger at the nucleotide level than the amino acid level. This primarily corresponds to synonymous nucleotide differences.

	Requires Fewest Changes		Requires Most Changes	
Tree	Nucleotides	Amino Acids	Nucleotides	Amino Acids
Standard Tree	6	1	25	13
Rhinolophoidea with Vespertilioniformes	5	2	23	10
Rhinolophoidea with Odontoceti	16	9	81	8

even within a single gene or protein, such as the prestin protein, phylogenetic trees are only capable of representing one signal, and thus, phylogenetic inference programs attempt to choose the signal with the greatest strength. This is not an accurate reflection of the data. Instead, we need a theory that can accommodate these conflicting signals.

3. A DEPENDENCY GRAPH MODEL OF AMINO ACID SEQUENCES

Such conflicting signals are expected under the dependency graph hypothesis. Recall that this hypothesis proposes that genomes were constructed by drawing on a number of modules rather than adapted from a single common ancestor. Consequently, different parts of the genome will show different patterns of similarities depending on which modules influenced that section of the genome. Likewise, different parts of a gene or protein would be expected to be influenced by different modules, and thus, show different signals of similarity. Our prior work [1] developed the dependency graph model in terms of the binary presence or absence of particular gene families. This was done primarily because it was the simplest and most straightforward way to explain the dependency graph model. We can extend that model by allowing modules not only to add or remove genes or proteins but also to modify them.

How would this apply to the particular case of echolocation and the prestin protein? The obvious approach might be to postulate the existence of an echolocation module. However, when we considered the sequences in Section 2 we found that there is not a strong signal of similarity shared among all echolocating prestins. Rather, we found that Rhinolophoidea's prestin is clearly similar both to Odontoceti's prestin and to Vespertilioniformes' prestin, but in different ways. Rhinolophoidea's prestin does not share the same set of similarities with Odontoceti's prestin that it does with Vespertilioniformes' prestin. Consequently, it makes sense to postulate two different modules, which we will call "Echolocation A" and "Echolocation B." We term them echolocation modules because we postulate that each represent a distinct optimization applied to prestin to facilitate echolocation.

Consider a possible dependency graph of the bat and cetacean clades depicted in Figure 3. In this graph, the Pteropodidae and Rhinolophoidea are related because they share a common module: Pteropodiformes. However, Rhinolophoidea is also related to Vespertilioniformes because it shares the "Echolocation A" module. Rhinolophoidea and Odontoceti share an "Echolocation B" module. This graph captures different relationships and can thus explain similarities among traditional evolutionary clades as well as between the microbats and among all the echolocating species.

Essentially, we can think of a module as a list of changes to the amino acid sequence. Each module merges together all of the changes from the modules it depends on before adding its own changes. For example, in Figure 3, the Echolocation A module inherits all of the changes made in Chiroptera and passes these them on to Rhinolophoidea and Vespertilioniformes. Pteropodiformes also inherits the changes from Chiroptera, and so Rhinolophoidea inherits two copies of the Chiroptera changes. However, because these are the same changes they do not conflict, and there is no problem.

Within the dependency graph model, every protein must be introduced by a module that defines the archetypical amino acid sequence for that protein. All other modules which modify that protein must depend on the module that introduced it. In consequence, as long as we are considering a single protein (as is the case in *AminoGraph*), there will always be one root module which introduces the protein and all other modules will have at least one dependency. This can be seen in Figure 3, in which all module dependencies trace back to Mammalia.

However, what happens if there is a conflict? What happens if a module has two dependencies each of which contain instructions to modify the same amino acid? Which change will actually apply? In software engineering, such conflicts are best avoided, and so, we simply

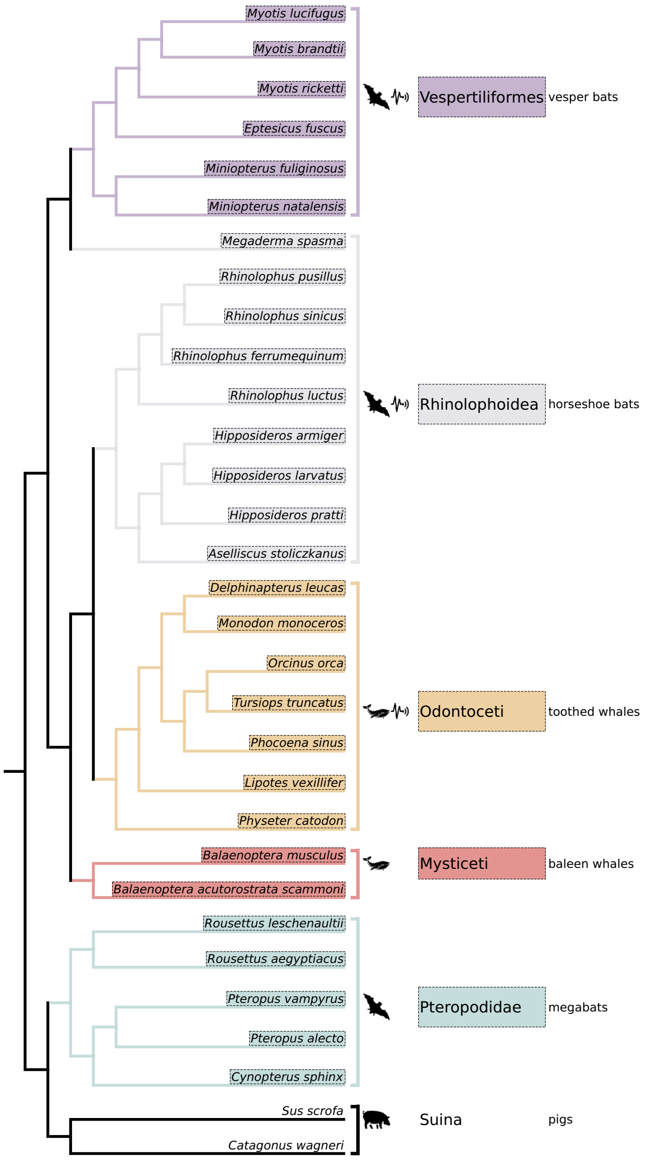

Figure 2: The subset of the phylogenetic tree inferred by Clustal W based on an amino acid alignment of prestin in 134 mammalian species. The subset containing the cetaceans and bats is shown. Pig-like animals are also included because the inference placed them within the bats and cetaceans. The icons and colors used for each clade correspond to Figure 1 for easier comparison. The echolocating clades have been grouped together despite being distantly related according to evolutionary theory. **doi:** 10.5048/BIO-C.2023.1.f2

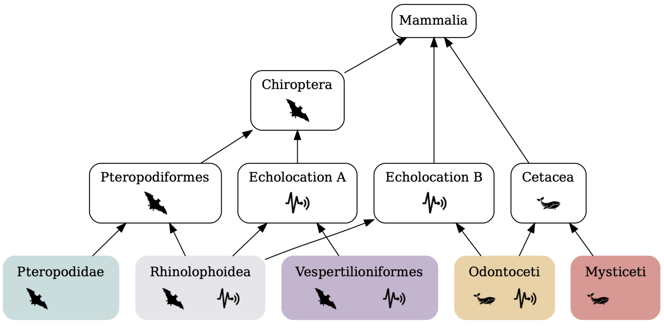

Figure 3: A hypothesized dependency graph relating the bats and cetaceans. The icons and colors used for each clade correspond to those in previous figures for easier comparison. Rather than a simple tree structure, this graph includes two echolocation modules which transcend a tree structure. The dependency graph model proposes that the species are related by this structure, explaining conflicting phylogenetic signals in the amino acid sequence. **doi:** 10.5048/BIO-C.2023.1.f3

forbid them in the model. This avoids scenarios in which the order of application of modules might change the outcome. However, there is an important exception. A module may override changes found in that module's dependencies. For example, a change made in "Echolocation A" may override a change made in Chiroptera. This follows common practice in software engineering, where a more specific version is allowed to override a more general version.

Efficiently tracking whether a given change counts as an override is difficult. Accordingly, *AminoGraph* uses an approximation. Each change has a "depth" that is incremented by one every time it is overridden. Changes with more depth are allowed to override changes with less depth. In cases in which a more derived module alters an inherited change, the depth will, by definition, be increased and thus have higher precedence than the changes it overrides.

AminoGraph takes any amino acid alignment as input and attempts to infer a dependency graph that best fits the data. It does this by exploring various graphs, looking for the one with the best fit to the data by drawing on Bayesian inference. In particular, it evaluates a prior over possible graph structures and a likelihood of the observed amino acid alignment given those structures. The prior is a probability distribution defining how probable different dependency graph structures are considered to be before looking at the evidence from the amino acid alignment. It is defined so that simpler structures have higher probability, thus favoring more parsimonious explanations. The likelihood is the probability of obtaining the observed amino acid sequences given the particular structure. The total fit to the data is the product of the likelihood and the prior, incorporating both the prior probability of the structure and the likelihood given that structure. See the appendix for the details of the formulas used to evaluate the prior and the likelihood.

4. RESULTS

4.1 Approach

We ran *AminoGraph* on a variety of amino acid alignments. When run, *AminoGraph* outputs a graph representing the structure that it infers to be present in the alignment. We classify these graphs into three different topologies. The simplest topology is a star topology, which models the alignment as the product of a single original sequence where each observed sequence in the alignment is an independently randomly modified version of that original sequence. In this topology, the different sequences have no relationship with each other besides being derived from the same original archetypical form. This is the result we expect when there is no structure in the data. The tree topology corresponds to a phylogeny from the theory of common descent. There is an ancestral sequence which undergoes a branching process, producing a tree. We expect this topology when the data are generated by something like common descent. The graph topology corresponds to the dependency graph model; We expect this topology when the alignment was actually produced by something like the model developed in this paper.

4.2 *AminoGraph* on Generated Amino Acid Alignments

As a first check to see if the results are realistic, we generated an amino acid sequence alignment following the dependency graph depicted in Figure 3. If the tool is unable to infer a dependency graph from data generated this way, this would indicate that it does not work as intended. The root, Mammalia, began with an amino acid sequence where each position was randomly selected uniformly from the set of all amino acids and a gap. For each node, five positions were assigned a random character, i.e. amino acid or a gap. Under each leaf, we introduced four new leafs corresponding to four individual species which evolved from the original ancestral type. In each case, we assume that the original type split into two species, and each of those two species split again to form a total of four species. For each split we, as for the nodes, applied five random amino acid changes. Our goal was to determine how well *AminoGraph* reconstructs the dependency graph from the resulting amino acid alignment.

Figure 4 depicts the resulting dependency graph. It closely resembles the dependency graph in Figure 3 upon which the generative process is based. It correctly infers the four species tree which was placed under each leaf in the original graph. The one error is that it infers Cetacea to depend on Chiroptera instead of inferring that Cetacea and Chiroptera both depend on a common module, Mammalia. Postulating that Cetacea depends on Chiroptera, or that Chiroptera depends on Cetacea, or that Chiroptera and Cetacea both depend on Mammalia all work approximately equally well to account for the similarities and differences in these two groups. However, postulating the existence of a Mammalia module is less parsimonious, and thus, *AminoGraph* prefers one of the other explanations. *AminoGraph* is able to correctly infer the rest of the structure including both echolocation modules.

Table 3: Results for random sequence alignments. Results include a variety of combinations of sequence length and number of sequences. Randomly generated alignments are consistently classified as a star phylogeny because they have no structure.

Sequences	Length	Topology	Probability (bits)
10	50	Star	2,201.6
10	100	Star	4,323.6
10	200	Star	8,560.2
10	500	Star	21,210.6
10	800	Star	34,078.1
50	50	Star	10,942.8
50	100	Star	21,607.9
50	200	Star	42,855.2
50	500	Star	107,158.5
50	800	Star	170,623.0
100	50	Star	21,996.7
100	100	Star	43,442.1
100	200	Star	86,054.4
100	500	Star	213,615.9
100	800	Star	341,551.5
150	50	Star	32,868.0
150	100	Star	64,882.5
150	200	Star	128,882.4
150	500	Star	320,592.0
150	800	Star	512,182.9
200	50	Star	43,942.4
200	100	Star	86,612.9
200	200	Star	172,063.7
200	500	Star	427,593.8
200	800	Star	682,064.4

4.3 *AminoGraph* on Random Amino Acid Alignments

As a second realism check, we ran *AminoGraph* on random data to determine whether or not *AminoGraph* infers structure where none exists. We generated a selection of random amino acid sequences of varying sizes and analyzed them. The results are shown in Table 3.

In every case, *AminoGraph* favored a star topology,

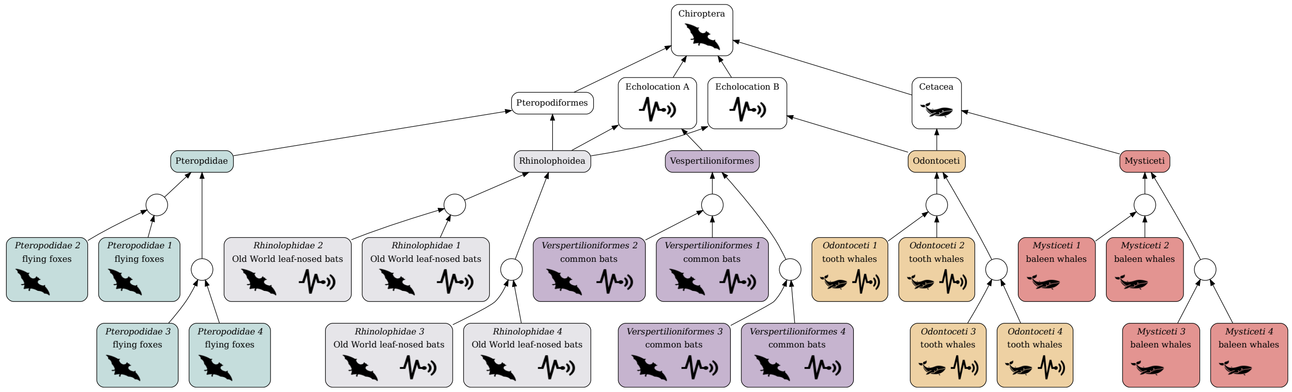

Figure 4: The dependency graph inferred by *AminoGraph* from an amino acid alignment generated following Figure 3. It closely resembles the dependency graph in Figure 3. *AminoGraph* correctly inferred the four leaf trees which were added in the generation process. The one difference between the generative process and the inferred graph is that Cetacea is inferred to depend on Chiroptera instead of both Chiroptera and Cetacea depending on Mammalia. **doi:** 10.5048/BIO-C.2023.1.f4

one in which there is a single root and all the sequences in the alignment are adapted versions of that root. Since there is no pattern to the random data, neither a tree nor a graph topology is a better explanation than the simple star topology. This is true across a variety of different numbers of sequences and sequence lengths. The probability column gives the number of bits required to explain the data. The numbers are large because the data are random, and thus, not well explained by the model.

4.4 *AminoGraph* on Pyvolve-Generated Sequences

As a third realism test, we evaluated how *AminoGraph* handles data that were produced by a process of simulated common descent. We employed an evolution simulator called Pyvolve [17] to generate amino acid alignments. We used the `ngesh` python module to generate a variety of trees of differing sizes and scale and used Pyvolve running under a number of different models to generate amino acid alignments. Table 4 presents the results.

In all but one case, *AminoGraph* inferred the alignment to best fit a tree. In the one exception, *AminoGraph* did not find enough evidence to support a tree and preferred a star phylogeny. The prior column gives the number of bits attributed to the graph prior for the inferred tree. The prior tends to increase as the number of sequences, sequence length, or median branch length increases. This is because as there are more sequences and changes to account for, *AminoGraph* infers more complex trees.

The likelihood column gives the number of bits attributed to the likelihood of the alignment given that particular tree. The probability gives the prior and the likelihood together. Note that the probabilities in this table are smaller for similar sized alignment than the ones in Table 3. This is because the data fit the tree model in this case. The model does explain the data.

The false bipartitions column gives the number bipartitions present in the inferred tree from *AminoGraph* but not present in the actual tree generated by `ngesh` and followed by `Pyvolve`. This is a simple metric for how similar the two trees are. It measures the number of groups identified by *AminoGraph* that were not actually present in the original tree. The extra nodes column gives the number of nodes inferred by *AminoGraph* aside from the ones dedicated to each sequence or the root. A node corresponds to a taxonomic category or ancestral species. We see consistently that only a small number of groups are inferred to exist by *AminoGraph* that were not present in the original tree. In many cases, it found zero false bipartitions, indicating that it did not infer any groups or clades that did not actually exist. This means that *AminoGraph* infers trees similar to those actually used in the simulation.

4.5 *AminoGraph* on TreeFam Alignments

We ran *AminoGraph* on fifty TreeFam [18] families randomly selected from those which had at least fifty sequences. The results are summarized in Table 5.

AminoGraph infers almost all of the sequences to have a graph topology. There are two exceptions. TF105771 is inferred to be a star topology. The alignment is unusual because one of the sequences is much longer than the rest, resulting in an alignment that is much longer than the most of the sequences. TF323869 is inferred to have a tree topology. It has the shortest median length—too short to infer a graph structure.

The extra nodes column gives the number of nodes in the graphs beyond the required nodes for each sequence

Table 4: Results for Pyvolve-generated sequences. Tests included a variety of sequences numbers, sequence lengths, mutational models, and median branch lengths. Sequences generated according to a model of common descent are consistently classified as either a tree or a star phylogeny. The sequence column gives the number of sequences in the generated alignment. It varies somewhat randomly because ngesh's random process does not guarantee a precise number of sequences. The length column gives the number of positions in the amino acid alignment. The model column gives the name of the substitution matrix used by Pyvolve to determine how probable different possible changes were. The median branch length column gives the median length of branches in the tree, obtained by scaling the trees generated by ngesh by different amounts. The small numbers in the false bipartitions column indicate that the tree inferred by the algorithm is similar to the one followed during the generation of the sequences. The extra nodes column indicates the degree to which the inferred tree deviated from a simple star phylogeny.

Sequences	Length	Model	Median Branch Length	Topology	Prior (bits)	Likelihood (bits)	Probability (bits)	False Bipartitions	Extra Nodes
25	50	JTT	0.00784	Tree	95.8	561.4	657.2	1	8
15	100	WAG	0.00294	Tree	18.6	549.2	567.7	0	1
16	200	LG	0.00152	Tree	62.9	1,127.0	1,189.9	0	7
14	500	DAYHOFF	0.00921	Tree	64.7	3,835.2	3,899.9	0	9
12	800	AB	0.00386	Tree	50.9	4,639.2	4,690.1	0	7
127	50	MTMAM	0.00173	Star	0.0	1,107.8	1,107.8	0	0
98	100	MTREV24	0.00588	Tree	640.9	2,918.0	3,558.9	1	51
123	200	JTT	0.00353	Tree	811.7	3,865.5	4,677.2	3	57
90	500	WAG	0.00176	Tree	599.3	5,123.3	5,722.5	0	52
113	800	LG	0.00719	Tree	878.4	23,571.8	24,450.1	0	91
175	50	DAYHOFF	0.00409	Tree	979.8	1,379.7	2,359.5	6	37
183	100	AB	0.00191	Tree	1,061.9	1,721.2	2,783.1	1	42
196	200	MTMAM	0.00674	Tree	1,512.6	9,794.3	11,307.0	7	113
204	500	MTREV24	0.00395	Tree	1,686.8	15,735.1	17,422.0	7	145
145	800	JTT	0.00140	Tree	1,079.6	10,745.2	11,824.8	1	89
275	50	WAG	0.00670	Tree	1,893.5	2,905.8	4,799.3	10	84
302	100	LG	0.00332	Tree	2,262.9	4,230.8	6,493.8	8	118
244	200	DAYHOFF	0.00167	Tree	1,597.9	3,602.6	5,200.4	3	68
300	500	AB	0.00797	Tree	2,602.6	38,920.7	41,523.3	0	202
270	800	MTMAM	0.00309	Tree	2,335.0	25,934.7	28,269.7	3	191
407	50	MTREV24	0.00184	Tree	1,879.8	1,673.3	3,553.1	5	22
400	100	JTT	0.00693	Tree	3,386.3	9,542.6	12,928.9	9	206
409	200	WAG	0.00353	Tree	3,514.4	11,407.2	14,921.7	5	220
396	500	LG	0.00166	Tree	3,338.5	14,236.4	17,574.9	7	202
355	800	DAYHOFF	0.00651	Tree	3,270.9	63,529.3	66,800.3	0	270

Table 5: Results for TreeFam alignments. In almost all cases, *AminoGraph* infers the data to be a graph rather than a tree. The extra dependencies column indicates the number of dependencies that would have to be removed in order to convert the graph into a tree. It is a measure of how far the graph diverges from a tree.

Treefam Family ID	Sequences	Alignment Length	Median Length	Topology	Prior (bits)	Likelihood (bits)	Probability (bits)	Extra Nodes	Extra Dependencies
TF105417	248	4307	1194	Graph	4,502.0	371,044.3	375,546.3	235	202
TF105709	96	1891	576	Graph	758.7	130,474.4	131,233.1	48	13
TF105771	189	5791	353	Star	0.0	392,263.6	392,263.6	0	0
TF106404	84	1229	571	Graph	843.6	89,746.7	90,590.2	57	26
TF300182	96	268	98	Graph	622.2	16,572.9	17,195.0	40	3
TF300659	166	1338	382	Graph	1,728.9	79,838.7	81,567.5	92	47
TF300677	102	733	499	Graph	1,166.6	67,926.5	69,093.2	73	42
TF300784	121	1670	538	Graph	1,202.1	131,447.4	132,649.5	75	31
TF312818	133	1077	524	Graph	1,517.9	123,650.8	125,168.7	95	47
TF313175	90	1575	327	Graph	682.0	79,301.3	79,983.3	43	11
TF313459	161	984	205	Graph	1,601.0	60,767.0	62,368.0	96	35
TF313797	304	960	328	Graph	3,843.5	123,180.4	127,023.9	199	106
TF313998	58	523	403	Graph	367.6	39,000.5	39,368.1	20	8
TF314814	115	594	140	Graph	751.5	22,781.1	23,532.5	41	5
TF314964	102	1797	792	Graph	1,188.3	137,109.2	138,297.5	67	48
TF315172	114	318	134	Graph	756.2	20,766.3	21,522.5	41	6
TF315217	167	3296	416	Graph	2,016.3	156,477.3	158,493.7	108	69
TF316050	123	937	338	Graph	980.6	98,008.3	98,988.9	59	14
TF316508	80	923	228	Graph	531.2	51,710.9	52,242.2	39	3
TF317588	108	591	176	Graph	938.0	27,602.9	28,540.9	52	23
TF318932	226	1598	107	Graph	1,815.5	90,108.5	91,924.0	97	13
TF319487	89	173	71	Graph	539.1	7,267.7	7,806.8	28	5
TF319889	65	627	279	Graph	526.9	24,019.9	24,546.8	34	14
TF320752	91	2658	882	Graph	960.6	205,192.5	206,153.1	52	38
TF321860	89	1104	238	Graph	690.4	48,441.7	49,132.0	44	12
TF323735	86	1626	446	Graph	840.9	70,909.6	71,750.5	52	27
TF323838	64	632	189	Graph	415.6	38,172.7	38,588.3	28	5
TF323869	58	308	58	Tree	284.7	6,556.6	6,841.4	19	0
TF324074	77	8683	875	Graph	669.7	202,871.3	203,541.1	34	24
TF324175	86	2379	1206	Graph	1,047.4	180,813.2	181,860.6	66	44
TF324238	88	1111	618	Graph	870.9	101,823.1	102,694.0	54	28
TF324402	57	156	79	Graph	313.7	9,032.2	9,346.0	22	1
TF324417	66	1266	788	Graph	540.3	100,882.6	101,422.9	34	15
TF324441	235	1910	532	Graph	2,535.9	310,406.0	312,941.9	138	57
TF324883	80	479	143	Graph	517.5	26,171.1	26,688.6	34	4
TF325196	148	4961	375	Graph	1,345.0	233,454.3	234,799.3	89	21
TF328358	204	1858	533	Graph	2,114.4	200,042.5	202,156.9	111	50
TF329158	167	1334	343	Graph	1,530.1	166,481.9	168,012.0	81	31
TF331604	102	1010	696	Graph	1,002.4	109,377.5	110,380.0	60	30
TF332303	56	507	450	Graph	399.9	26,415.1	26,815.0	26	9
TF332900	87	3140	921	Graph	953.5	150,829.4	151,783.0	57	37
TF333215	57	424	254	Graph	362.5	33,026.3	33,388.8	28	3
TF336515	110	642	463	Graph	1,025.7	52,946.8	53,972.5	60	27
TF339438	51	228	139	Graph	317.4	12,195.2	12,512.6	21	5
TF342033	95	955	489	Graph	1,343.1	85,632.6	86,975.7	73	69
TF342861	63	891	385	Graph	483.6	78,466.3	78,949.9	36	9
TF350735	371	2291	279	Graph	7,269.9	223,416.8	230,686.7	376	306
TF350893	86	1854	710	Graph	993.4	112,652.7	113,646.1	73	34
TF351335	257	1820	321	Graph	3,647.2	153,643.1	157,290.3	180	133
TF352582	105	1126	424	Graph	918.6	107,282.0	108,200.7	61	18

and the root. The extra dependencies column gives the number of dependencies beyond the one required per non-root node. It is the number of dependencies that would have be pruned in order for the graph to become a tree, and therefore a measure of how far from a tree the graph is.

4.6 *AminoGraph* and Prestin Alignment

AminoGraph infers the prestin alignment, discussed previously in this paper, to be best explained by a dependency graph. Figure 5 depicts the dependency graph restricted to the nodes related to the bats and cetaceans, resembling the hypothesized dependency graph in Figure 3. There is a Chiroptera or bat module that all the various bat species depend on. There is a Cetacea module that all the cetaceans depend on. The five major groups each show up in a cluster in this graph.

Megaderma spasma is grouped with Vespertilioniformes instead of with Rhinolophoidea. The same misplacement was found in the inferred phylogenetic tree in Figure 2. The prestin sequence in *Megaderma spasma* more closely resembles that of Vespertilioniformes than that of Pteropodidae. Thus, when considering only the sequence of prestin, it is grouped with those species.

There is an Echolocation A module, which all microbats depend on. However, there is no module corresponding to Pteropodiformes, the combination of megabats and those microbats thought to be more closely related to them. Recall that when we considered the sequence data, we found there was a very weak signal in the prestin to group these species together. As such, it makes sense that *AminoGraph* did not infer one here. The Echolocation B module is depended on both by the echolocating whales (Odontoceti) and by some of microbats (Rhinolophoidea). However, it is not depended on by Vespertilioniformes or the microbat module. This reflects the signal showing similarities between the toothed whales (Odontoceti) and one clade of microbats (Rhinolophoidea).

When we first started this research project, we expected to find a single echolocation module. This was based on an assumption that there were similarities across all of the echolocating clades of bats and cetaceans that explained the misplacement found in previous phylogenetic inferences [12, 13]. However, as we developed the *AminoGraph* tool, it refused to follow our preconceptions, instead following the data. As discussed, the data indicates three distinct signals of similarity and *AminoGraph* is able to detect and separate these three signals. Compare this with the conclusions that can be drawn from the phylogenetic inference depicted in Figure 2. The tree is simply not capable of capturing the complexity of the patterns of similarities that the dependency graph model can.

5. CONVERGENT EVOLUTION

It is clear that these amino acid sequences contain conflicting phylogenetic signals. Furthermore, these conflicts cannot be explained as being due to incongruence between the gene tree and the species tree. In most cases, they cannot be explained via any sort of exotic evolutionary history. Indeed, in evolutionary terms, the “only remaining reason” [12] is convergent evolution under the influence of natural selection. Under this hypothesis, there are conflicting signals because of the combination of common descent and natural selection.

Convergent evolution due to natural selection is undoubtedly a real process that explains some biological similarities. For example, convergent evolution has been observed in the ongoing evolution of SARS-CoV-2 [19]. However, this is the ideal circumstance to enable convergent evolution: an enormous population size, small genome, high uniformity, and large selection effects. In the case of the evolution of complex lifeforms, such as mammals, we have small populations, large genomes, high diversity and small selection effects. It is unexpected that convergent evolution would apply to these situations. Indeed, the papers which published the molecular convergence in prestin describe it as surprising or unexpected [12, 13]. They invoke convergent evolution not because it is an expected outcome but because it the only remaining evolutionary option. Convergent evolution does not seem a viable account of a widespread pattern of conflicting phylogenetic signals.

Nevertheless, our purpose here is to develop the dependency graph model rather than to disprove the possibility of convergent evolution. We wish to demonstrate the viability of the dependency graph model as an account of the similarities and differences in amino acid alignments. Definitively disproving alternative accounts is beyond the scope of this paper. We have given some reasons to doubt the viability of selection-driven convergent evolution as an account, but leave to future research a fuller examination of the issue.

However, there is a common argument made for the convergent evolution explanation. An example can be found in one of the papers that postulated prestin convergence [12]:

> Indeed, the same misplacement of dolphin is observed in the prestin tree reconstructed with only nonsynonymous nucleotide substitutions (Figure S1B); but, when only synonymous substitutions are used, dolphin and cow are correctly grouped with 100% bootstrap support.

The idea is that natural selection can cause convergent evolution for nonsynonymous mutations, because they alter the amino acid sequence, and thus, the fitness of the protein. However, natural selection would not cause

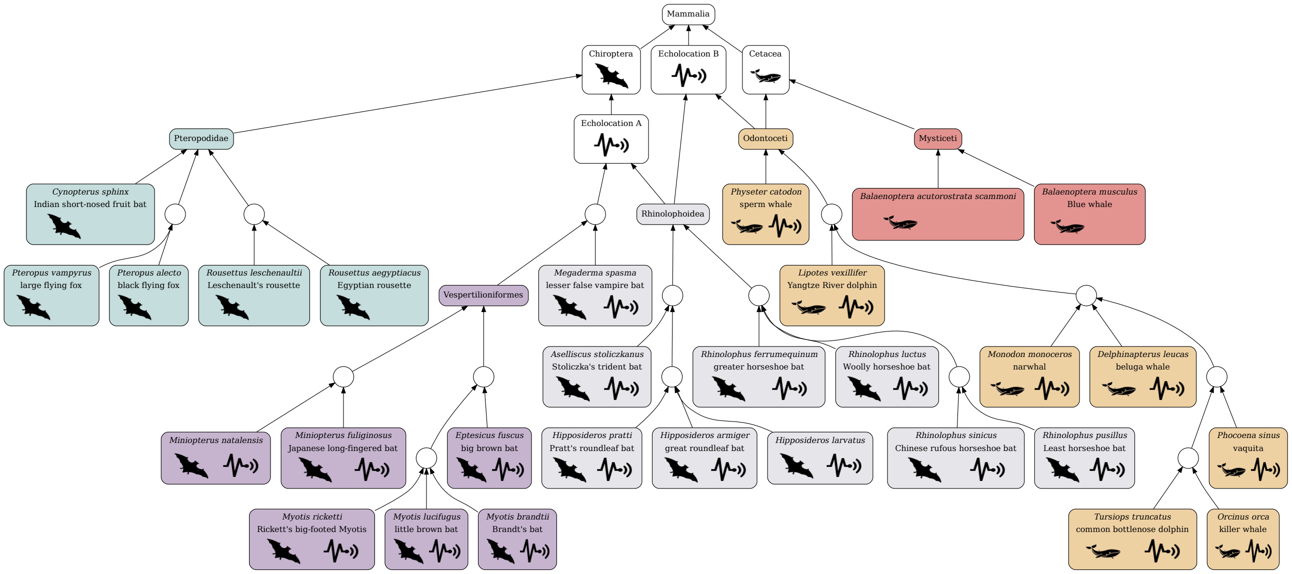

Figure 5: Inferred dependency graph from the prestin protein for bats and cetaceans. The icons and colors used for each clade correspond previous figures for easier comparison. The figure resembles a more detailed version of Figure 3. **doi:** 10.5048/BIO-C.2023.1.f5

convergent evolution for synonymous mutations because (it is argued) these changes have little to no effect on the fitness of the protein. As such, the hypothesis of convergent evolution explains why it is primarily among nonsynonymous mutations that the data conflict with the expectations of common descent.

However, we know that synonymous mutations affect proteins in a variety of ways [20–24]. As such, it is not that synonymous mutations have no effect, but the effects of synonymous mutations differ in degree and kind from the effects of nonsynonymous mutations. It is expected that different modules serving different purposes introduce mutations with different kinds of effects. In particular, an echolocation module has no reason to introduce synonymous mutations that are unlikely to optimize the protein for echolocation. Rather, we should expect an echolocation module to only modify the gene in ways that contribute to echolocation.

6. CONCLUSIONS

We have extended the dependency graph model to amino acid sequences. In so doing, we have offered an explanation for discordant phylogenies and conflicting phylogenetic signals. We have shown that data that are either randomly generated or produced by a simulated branching process do not exhibit these conflicting signals. However, as shown here with *prestin* sequences, real genetic data can have such conflicting signals. The new *AminoGraph* tool provides a way for users to explore the conflicting signals and potential dependency graph influences of amino acid sequences.

Our evidence for the correctness of our model is the various sequences that *AminoGraph* detects as exhibiting the structure expected based on a dependency graph. This is akin to many arguments for common descent that identify hierarchical signals in various datasets. We are doing essentially the same thing by showing that there is a dependency graph signal in these datasets. In fact, as depicted in Figure 6, the dependency graph signal is a refinement of the hierarchical signal. That is, cases of sequences that exhibit dependency graph signals will also exhibit a hierarchical signal. However, almost all sequences that exhibit a hierarchical signal would not also exhibit a dependency graph signal. Even data that deviate from a hierarchical signal would not tend to exhibit a dependency graph signal because the dependency graph signal requires that a pattern of amino acid substitutions appear in distinct groups. Finding this pattern is a successful prediction of the dependency graph model.

Ultimately, however, the dependency graph model needs to match and exceed the predictive power of common descent. That includes all predictions from all fields of biology. Roughly, the dependency graph model would suggest that these predictions were possible using common descent only because the tree of life is an approximation to the most significant modules in the dependency graph. As such, the dependency graph still underwrites these predictions while also better predicting and explaining homoplasies. Nevertheless, the dependency graph is an immature model and requires more development

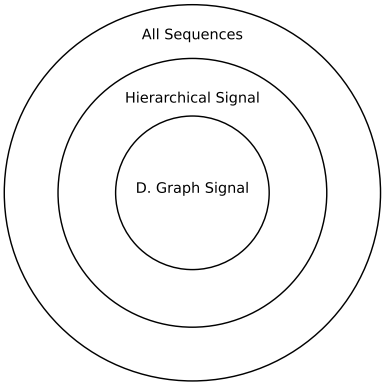

Figure 6: A Venn diagram depicting the relationship between all amino acid sequences, amino acid sequences exhibiting a hierarchical signal, and amino acid sequences exhibiting a dependency graph signal. All sequences that exhibit dependency graph signals will also exhibit a hierarchical signal, but not *vice versa*. doi: 10.5048/BIO-C.2023.1.f6

before it can be claimed as the best explanation over common descent.

For this paper, we put forward an extension of the dependency graph model to amino acid sequences. This demonstrates the basic viability of this model to explain sequence data and not merely the presence or absence of gene families. We have shown that there is a signal in the data consistent with a dependency graph model. Any other model of amino acid sequences must explain this signal. Additional research is required to evaluate and develop this model as a full fledged alternative to common descent.

7. APPENDIX: PROBABILISTIC MODEL

AminoGraph uses Bayesian reasoning to attempt to infer a dependency graph from an amino acid alignment. Bayes theorem applied to dependency graphs states that:

$$\Pr[G|D] = \frac{\Pr[D|G]\Pr[G]}{\Pr[D]} \tag{1}$$

where $\Pr[D]$ is the probability of the amino acid alignment, $\Pr[G]$ is the prior probability of a particular dependency graph, $\Pr[D|G]$ is the probability of the amino acid given a particular dependency graph, and $\Pr[G|D]$ is the probability of the graph given the amino acid alignment. We wish to know that graph which has the highest probability given the supplied amino acid alignment, $\Pr[G|D]$. By Bayes theorem, this is the graph which maximizes $\Pr[D|G]\Pr[G]$.

The graph prior $\Pr[G]$ is the prior probability of a particular dependency graph. Let s be the number of sequences in the alignment being investigated—this is taken as a given. Let $n \geq s+1$ be the number of nodes in the graph. It is at least one node for each sequence and the root. We assume the number of additional nodes is drawn from a geometric distribution. Taking the prior of the parameter of geometric distribution as uniform between 0 and 1, we obtain:

$$\Pr[n] = \int_0^1 (1-p)^{n-s} p \, dp = \beta(n-s, 2) \tag{2}$$

The extra nodes added to the root and sequence nodes are unlabeled. This means that for every graph containing extra $n-s$ nodes there are $(n-s)!$ other graphs which are the same, except the extra nodes are recorded in a different order. We must factor these equivalent graphs into the probability of the graph being studied by multiplying the prior by $(n-s)!$.

The root has no dependencies by definition. Each node other than the root has at least one dependency. We take the number of additional dependencies for each node to be taken from a geometric distribution. Each dependency must be to one of the inferred nodes in the graph and not to one that represent a sequence provided in the alignment. The probability of the specific number of dependencies of all nodes can be expressed as:

$$\begin{aligned}\int_0^1 \prod_i \frac{1}{n-s}\left(\frac{1-p}{n-s}\right)^{P_i-1} p \, dp &= \frac{\int_0^1 (1-p)^{d-n-1} p^{n-1} \, dp}{(n-s)^d} \\ &= \frac{\beta(d-n, n)}{(n-s)^d}\end{aligned}$$

where i is a non-root node in the graph, p is the parameter of the geometric distribution, P_i is the number of dependencies of node i, and d is the total number of dependency relationships in the graph.

For non-sequence nodes, we take the number of dependencies to also be taken from a geometric distribution. The prior thus comes to the following:

$$\Pr[G] = \frac{\beta(n-s,2)(n-s)!\beta(d-n,n)}{(n-s)^d} \tag{3}$$

However, this prior assigns probability to invalid graphs, such as one containing cycles where a dependency depends on itself. Ideally, we would compute the probability of the graph given that the graph is valid. This may expressed as $\Pr[G|V]$, where V means a valid graph is generated. However, we wish to maximize the quantity $\Pr[D|G]\Pr[G|V]$, which is equivalent to $\frac{\Pr[D|G]\Pr[G]}{\Pr[V]}$ for all valid graphs. The $\Pr[V]$ does not depend on G, and thus, the G that maximizes $\Pr[D|G]\Pr[G]$ also maximizes $\Pr[D|G]\Pr[G|V]$.

Two restricted subsets of possible graphs are of interest: stars and trees. In a star phylogeny, there is only one root and the sequences, and each sequence depends directly on the root. In a tree phylogeny, each non-root node has one dependency. *AminoGraph* automatically determines which subset a graph falls into and adjusts the prior to be specific to that subset. There is only one possible star phylogeny, so its prior is always one. For a tree topology, it is impossible for a node to have extra dependencies, so the associated probability is taken to be 1 instead of $\beta(d-n,n)$.

The likelihood $\Pr[D|G]$ is the probability of provided amino acid alignment given the dependency graph. *AminoGraph* takes an amino-acid model in Paml triangular format. This provides a matrix with element T_{ij} providing the transition rate from amino acid i to amino acid j. It also provides a vector I_i providing the frequency of the various amino acids, normalized to sum to one to form a valid probability distribution.

For the root node, we use a geometric distribution over possible lengths of the initial sequence.

$$\int_0^1 (1-p)^l p\, dp = \beta(l+1,2) \tag{4}$$

Additionally, we must incorporate the probability of each amino acid in the initial sequence. Thus, the probability of the root node is $\beta(|s|+1,2)\prod_j I_{s_j}$, where s is the sequence of the root node and j is the valid indices of that sequence.

For other nodes, we determine the inherited state of the position: whether it be empty or a specific amino acid. There are two additional probabilities of interest.

- p_i is the probability of an insertion
- p_d is the probability of a deletion

If the amino acid is present in the sequence (i.e. it is not a gap and it is not deleted), the probability is $(1-p_d)e^{Tl}_{ij}$, where l is a parameter indicating the amount of evolution expected in a node, and i is an index corresponding to the inherited amino acid, and j is an index corresponding to the amino acid actually observed. The parameter l is optimized during the search process to maximize the probability of the graph. The probability of a deletion is p_d. The probability of an insertion is $p_i I_j$, where j is the amino acid inserted. Potential insertions exist before and after each present amino-acid position and each has a probability of $(1-p_i)$.

The probability of each node can be expressed as follows:

$$p_i^{a_i}(1-p_i)^{b_i} p_d^{a_d}(1-p_d)^{b_d}\gamma \tag{5}$$

The *a*s and *b*s are computed by counting the cases described above, and γ is computed by looking up the matrix T and vector I. Each $p^a(1-p)^b$ can be aggregated to the graph level where we can take a uniform prior over possible values of p.

$$\int_0^1 p^a(1-p)^b = \beta(a+1,b+1) \tag{6}$$

Thus, the full likelihood may be computed by multiplying the various beta functions and product of the γ across all nodes.

In principle, it would be best to compute the likelihood by summing over all possible assignments of states to all non-sequence nodes. However, this is impractical. As such, *AminoGraph* instead computes the likelihood of one particular assignment of states and seeks to identify the most probable state.

1. Ewert W (2018) The Dependency Graph of Life. BIO-Complexity 2018(3):1–27. doi:10.5048/BIO-C.2018.3
2. Theobald DL (2010) A formal test of the theory of universal common ancestry. Nature 465(7295):219–222. doi:10.1038/nature09014
3. Penny D, Foulds LR, Hendy MD (1982) Testing the theory of evolution by comparing phylogenetic trees constructed from five different protein sequences. Nature 297(5863):197–200. doi:10.1038/297197a0
4. White WTJ, Zhong B, Penny D (2013) Beyond Reasonable Doubt: Evolution from DNA Sequences. PLoS ONE 8(8):e69924. doi:10.1371/journal.pone.0069924
5. Baum DA, Ané C, Larget B, Solís-Lemus C, Ho LST, Boone P, Drummond CP, Bontrager M, Hunter SJ, Saucier W (2016) Statistical evidence for common ancestry: Application to primates. Evolution 70(6):1354–1363. doi:10.1111/evo.12934
6. Rokas A, King N, Finnerty J, Carroll SB (2003) Conflicting phylogenetic signals at the base of the metazoan tree. Evolution and Development 5(4):346–359. doi:10.1046/j.1525-142X.2003.03042.x
7. Rokas A, Williams BL, King N, Carroll SB (2003) Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature 425(6960):798–804. doi:10.1038/nature02053
8. Jeffroy O, Brinkmann H, Delsuc F, Philippe H (2006) Phylogenomics: The beginning of incongruence? Trends in Genetics 22(4):225–231. doi:10.1016/j.tig.2006.02.003
9. Smith SA, Moore MJ, Brown JW, Yang Y (2015) Analysis of phylogenomic datasets reveals conflict, concordance, and gene duplications with examples from animals and plants. BMC Evolutionary Biology 15(1):1–15. doi:10.1186/s12862-015-0423-0
10. Nosenko T, Schreiber F, Adamska M, Adamski M, Eitel M, Hammel J, Maldonado M, Müller WE, Nickel M, Schierwater B, Vacelet J, Wiens M, Wörheide G (2013) Deep metazoan phylogeny: When different genes tell different stories. Molecular Phylogenetics and Evolution 67(1):223–233. doi:10.1016/j.ympev.2013.01.010
11. Maddison WP (1997) Gene trees in species trees. Systematic Biology 46(3):523–536. doi:10.1093/sysbio/46.3.523
12. Li Y, Liu Z, Shi P, Zhang J (2010) The hearing gene Prestin unites echolocating bats and whales. Current biology : CB 20(2):R55–6. doi:10.1016/j.cub.2009.11.042
13. Liu Y, Cotton JA, Shen B, Han X, Rossiter SJ, Zhang S (2010) Convergent sequence evolution between echolocating bats and dolphins. Current Biology 20(2):1–3. doi:10.1016/j.cub.2009.11.058
14. Herrero J, Muffato M, Beal K, Fitzgerald S, Gordon L,

Pignatelli M, Vilella AJ, Searle SMJ, Amode R, Brent S, Spooner W, Kulesha E, Yates A, Flicek P (2016) Ensembl comparative genomics resources. Database 2016:1–17. doi:10.1093/database/bav096

15. Zdobnov EM, Tegenfeldt F, Kuznetsov D, Waterhouse RM, Simão FA, Ioannidis P, Seppey M, Loetscher A, Kriventseva EV (2017) OrthoDB v9.1: Cataloging evolutionary and functional annotations for animal, fungal, plant, archaeal, bacterial and viral orthologs. Nucleic Acids Research 45(D1):D744–D749. doi:10.1093/nar/gkw1119
16. Larkin MA, Blackshields G, Brown NP, Chenna R, Mcgettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG (2007) Clustal W and Clustal X version 2.0. Bioinformatics 23(21):2947–2948. doi:10.1093/bioinformatics/btm404
17. Spielman SJ, Wilke CO (2015) Pyvolve: A flexible python module for simulating sequences along phylogenies. PLoS ONE 10(9):1–7. doi:10.1371/journal.pone.0139047
18. Ruan J, Li H, Chen Z, Coghlan A, Coin LJM, Guo Y, Heacute;riché JK, Hu Y, Kristiansen K, Li R, Liu T, Moses A, Qin J, Vang S, Vilella AJ, Ureta-Vidal A, Bolund L, Wang J, Durbin R (2008) TreeFam: 2008 Update. Nucleic Acids Research 36(SUPPL. 1):735–740. doi:10.1093/nar/gkm1005
19. Zahradník J, Nunvar J, Schreiber G (2022) Perspectives: SARS-CoV-2 Spike Convergent Evolution as a Guide to Explore Adaptive Advantage. Frontiers in Cellular and Infection Microbiology 12:1–7. doi:10.3389/fcimb.2022.748948
20. Parmley JL, Hurst LD (2007) How do synonymous mutations affect fitness? BioEssays 29(6):515–519. doi:10.1002/bies.20592
21. Chamary JV, Hurst LD (2005) Evidence for selection on synonymous mutations affecting stability of mRNA secondary structure in mammals. Genome biology 6(9):1–12. doi:10.1186/gb-2005-6-9-r75
22. Quax TE, Claassens NJ, Söll D, van der Oost J (2015) Codon Bias as a Means to Fine-Tune Gene Expression. Molecular Cell 59(2):149–161. doi:10.1016/j.molcel.2015.05.035
23. Rauscher R, Ignatova Z (2018) Timing during translation matters: Synonymous mutations in human pathologies influence protein folding and function. Biochemical Society Transactions 46(4):937–944. doi:10.1042/BST20170422
24. Lebeuf-Taylor E, McCloskey N, Bailey SF, Hinz A, Kassen R (2019) The distribution of fitness effects among synonymous mutations in a gene under directional selection. eLife 8(e45952):1–16. doi:10.7554/eLife.45952.001