Towards the true tree:
Bioinformatic approaches in the phylogenetics and molecular evolution of the Endopterygota

Student: Douglas Chesters

Supervisors: Alfried Vogler and Donald Quicke

Imperial College, Division of Biology

A thesis submitted to Imperial College for the degree of Doctor of Philosophy

I hereby declare that all work presented here is my own. Work of others is appropriately referenced.

Signed:

Date:

Abstract

In this thesis, I use bioinformatic approaches to address new and existing issues surrounding large-scale phylogenetic analysis. A phylogenetic analysis pipeline is developed to aid an investigation of the suitability of integrating Cytochrome Oxidase Subunit 1 (cox1) into phylogenetic supermatrices. In the first two chapters I assess the effect of varying cox1 sample size within a large variable phylogenetic context. As well as intuitive results on increased quality with greater taxon sampling, there are clear monophyly patters relating to local taxonomic sampling. Specifically, more monophyletic resampled taxa in cases when fewer consubfamilials are represented, with a tendency for these to remain unchanged in the degree of monophyly when rarefied. Sampling analyses are extended in chapter two using a mined Scarabaeoidea multilocus dataset, where taxa from given loci are used to improve existing matrices. Improvement in phylogenetic signal is best achieved by targeting cox1 to existing taxa, which suggests minimum parameters for cox1 adoption in large-scale phylogenetics.

In chapter 3 I address recently-arisen issues related to phyloinformatic analysis of sequence-delineated matrices. There is ongoing work on setting species boundaries by sequence variation alone, but incongruence results in methodological issues upon integrating multiple loci delineated in this way.

In the final chapter I assess the impact of heterogeneous substitution rates on large scale cox1 datasets. Although the number of heterogeneous sites in Coleoptera cox1 is substantial, their presence is found to be beneficial, as their removal negatively impacts the ability of the alignment to generate the 'known' topology. The homoplasy and heterogeneous characteristics of cox1 have not substantially impacted its utility, thus the cox1 datasets have potential to play a substantial role in the tree-of-life.

Table of Contents

Declaration of content 2

Abstract 3

Table of Contents 4

List of Tables 7

List of Figures 8

Acknowledgments 9

Chapter one – Introduction 10

- 1.1: Data Requirements for Phylogenetics 10
- 1.2: DNA Barcoding 11
- 1.3: Consolidating Data 12
- 1.4: Multiple Sequence Alignment 15
- 1.5: Building Trees from Multiple Loci 16
- 1.6: Determination of Tree Accuracy 17
- 1.7: Factors Influencing the Accuracy of Inferred Trees 18
- 1.8: Implementation; the Bioinformatic Approach 20
- 1.9: Synopsis of Content 22

Chapter two – Taxon Sampling and Braconid Wasp cox1 Phylogeny 23

- 2.1: Introduction 23
- 2.2: Methods 26
- 2.3: Results 32
- 2.4: Discussion 43

Chapter three – Improving Existing Supermatrices With cox1 48
3.1: Introduction 48
3.2: Methods 51
3.3: Results 57
3.4: Discussion 63

Chapter four – Supermatrices from Sequence Delineated Loci 67
4.1: Introduction 67
4.2: Methods 70
4.3: Results 76
4.4: Discussion 86

Chapter five – Site-specific rate variation in cox1 90
5.1: Introduction 90
5.2: Methods 94
5.3: Results 99
5.4: Discussion 107

Chapter six – General Discussion and Conclusions 111
6.1: The Analysis Framework; HIP-DB 111
6.2: Resolving the Tree of Life With cox1? 112
6.3: Limitations 115
6.4: Conclusions 116

References 117

Appendices 132
1: Key Subroutines
1.1: Taxstring generation 132
1.2: Database searching 138
1.3: Tree Pruning Algorithm 141

1.4: Taxonomic Consistency Calculation 146
1.5: Tree-building Subroutine, RAxML Wrapper 149
1.6: Geographic Website Scraping 152

List of Tables

2.1: Monophyly and Congruence Scores 32

2.2: Subfamily Summary 35

2.3: Chi Squared Results 37

3.1: Subfamily Counts for Various Loci 58

3.2: Mean Taxonomic Consistency 58

3.3: Statistical Results 60

4.1: Taxcode Generation Example 51

4.2: Species Assignment to Unidentified Sequences 78

4.3: Congruence in Sequence Clustering Between Loci 79

5.1: Mean Values Retrieved from ML Analysis 102

5.2: Taxonomic RI Scores 103

List of Figures

2.1: Ensemble taxonomic congruence over search time 28
2.2: Comparing tRI and tSD scores 29
2.3: Illustrating Taxon Pruning 31
2.4: Plot of change in congruence according to sample size 39
2.5: Proportion of robust taxa 40
3.1: HIP-DB flowchart 51
3.2: Illustrating Geographic Assignment to Trees 53
3.3: Resampling Analysis Flowchart 56
3.4: Proportion of robust resamples 59
3.5: Plot of change in quality accorning to cox1 addition. 61
3.6: Plot of change in quality accorning to addition of other loci. 62
4.1: Illustrating a search for optimal combination 73
4.2: Scatter Plots of BlastClust Cutoffs. 77
4.3: Frequency histogram of taxonomic match scores 80
4.4: A scatter plot of total concatenate combination score vs time 81
4.5: Illustrating the relationship of clusters with taxa 83
4.6: Comparing Species Diversity between Supermatrices 84
4.7: ML trees from sequence/taxon delineated supermatrices 85
5.1: Illustrating the biasing form of heterotachy 97
5.2: Sliding window plot of variability 100
5.3: Significantly heterotachous sites 102
5.4: frequency histogram of p-values 106

Acknowledgments

Many people have been invaluable in various aspects of my PhD. In terms of the work presented here, I would like to thank Patrick Brock (for discussions laying the foundation of chapter four) and Toby Hunt (for introducing me to phyloinformatics procedures). I would also like to thank the many people I have collaborated with, particularly Deyan Ge, Chris Borrow and the NHM masters students.

I am very grateful of the distractions provided by, in particular, my Romanian & Slovakian cohabitees, and the MRes students of 2009.

Finally, I would not have achieved this without the support of my supervisors Alfried Vogler and Donald Quicke, and my colleagues at the NHM Molecular Lab. My deepest appreciation goes to all these people.

Chapter One: Introduction

Since Darwin's concept of representing species relatedness through genealogical trees, scientists have been steadily working towards assembling the tree of life. Genetic technologies developed in the 1970's and 1980's allowed the extraction of data which gave scientists great power to resolve the branching pattern of the tree. Since this time, the ability to extract this data has increased exponentially. While the study of tree of life is benefitting, the sheer amount of molecular data itself presents a range of additional issues in handling and analysis. But the integration of informatics techniques is proving highly successful in addressing these issues. Large scale phylogenetic trees utilise molecular sequences held in public databases, of which Genbank is the best known (http://www.ncbi.nlm.nih.gov/genbank/). Genetic information available on such databases include genome sequences (available only for a small number of species), and individual genes/gene fragments (some of which are available for a large number of species). Many different approaches to molecular phylogenetics exist, using different types of genetic data. But perhaps the most feasible approach to building the tree of life is that of combining datasets from a small number of loci which have been sampled (taxonomically) densely (e.g. Golobof et al., 2009).

1.1: Data Requirements for Phylogenetics

The growth of sequence data, whilst remarkable, does not alter the fact that only a small proportion of the genome is available for only a small fraction of the earth's species diversity, and only a subset of this data is appropriate for the task of tree-building. Are there features of this dataset which hinder efforts to build a phylogeny of sufficient quality? Decisions both on which data are sampled in sequencing efforts, and which are used for phylogenetics, have consequences for the constituency of the tree (of course), but also the likelihood of successfully creating an evolutionary model with good fit, and therefore the probability that the result inferred has any meaning or usefulness. Thus the important questions that need to be addressed are: what do we require from molecular data in order to assemble a reasonable tree of life, how do we physically sample genetic

diversity in order to attain this necessary data, and how can we tell whether our inferred results have the expected level of accuracy?

The homologous molecular sequences most useful for phylogenetic inference are those not only with a substantial number of characters, but characters that are able to differentiate between competing tree topologies (Wortley and Scotland, 2006). Sequences with phylogenetic signal are those giving a consistent (un-biased) tendency to a given phylogenetic hypothesis (Huelsenbeck and Bull, 1996). Different loci can vary substantially in these features (e.g. Burleigh and Mathews, 2007; Danforth et al., 2005; Rokas et al., 2002), and this is confounded as the signal may be useful in one context but not another, e.g. when the evolutionary scale is varied (Misof et al., 2001). Clearly loci are required that contain specific characteristics, but it may also be the case that a lower bound for the absolute number of characters exists to resolve a tree of a particular number of taxa (Wiens, 2003). The number of taxa sampled influences the susceptibility to biasing influences (Rannala et al., 1998), the fit of the data to the evolutionary model (Pollock and Bruno, 2000), and therefore the degree of error (Zwickl and Hillis, 2002). Still, taxon and locus sampling requirements are a topic of ongoing discussion (Heath et al., 2008).

In this thesis I address the question of locus and sample requirements in the two largest radiations of living animals, the Coleoptera (beetles) and Hymenoptera (wasps, bees and ants). I use two quite different underlying cox1 datasets to determine the sampling requirements and resulting phylogentics signal of this gene. In chapter two a well curated and densely sampled wasp dataset is used, contrasting the mined beetle dataset in chapter three. It is necessary to discover the phylogenetic properties and sampling requirements of cox1 in particular due to its commonality. Cox1 is the single most commonly sequenced locus, a trend that is only likely to increase due to DNA barcoding studies.

1.2: DNA Barcoding

As speciation history is reflected in the pattern of morphological traits used in taxonomy, this raises the possibility of using DNA sequence data to identify morphospecies. Quite recently, Paul Hebert and colleagues (2004), proposed this was feasible with a single

fragment of mtDNA, the method known as DNA barcoding. Excitement surrounding the proposal has resulted in the ambitious project to barcode the majority of life on earth (Savolainen et al., 2005). The use of a standardised gene fragment (cox1) for such an endeavor is clearly necessary if there is to be any hope of compiling such a densely sampled dataset (Caterino et al., 2000). Cox1 is a locus that has been long used for phylogenetic analysis in the Endopterygota (Howland and Hewitt, 1995; Lunt et al., 1996; Caterino et al., 2000. Beutel and Pohl, 2006; Hunt et al., 2007), which raises the possibility that the barcode databases may be co-opted for phylogenetic analysis. The characteristics and degree of variation of mtDNA and cox1 in particular make it more suited for resolution at the species level (Avise et al., 1987; Hebert et al., 2003), but mtDNA does not always recover an accurate species tree (e.g. Will and Rubinoff, 2004). This could be a failing of the simplistic clustering procedures used in barcoding studies (Will and Rubinoff, 2004), in which case the development of statistical approaches (Nielsen and Matz, 2006) may account for. But theory suggests that even correctly modeled mtDNA will be at discord with a species tree, resulting in a number of difficulties.

1.3: Consolidating Data

Data obtained from repositories needs to be consolidated into a form suitable for phylogenetic inference. This requires both the definition of the fundamental units, i.e. the terminals in the phylogenetic analysis, and the assignment of homologous characters (typically requiring the combination of multiple characters sets).The fundamental units of the tree of life are the species. Defining a species means both deciding what features characterise a species (a species concept), and then applying these criteria to delineate the taxa to be included in the phylogenetic analysis. For example species concepts have been developed which group individuals or populations based on reproductive isolation (Dobzhansky, 1970), a shared ecological niche (Van Valen, 1976), the possession of unique evolutionary tendencies (Wiley, 1978), or the sharing of unique diagnosable character states (Nixon and Wheeler, 1990). de Queiroz (2007) points out the commonality amongst these concepts, of separately evolving lineages (where lineages are a groups of connected populations), and suggests many ‘alternative’ concepts are simply

secondary by focusing on the properties of group that have undergone separate evolution. The distinction is useful in that these secondary properties may be used in isolation (from the unified concept), in the process of delimitation. In other words, speciation results in unique characteristic of a lineage, so the species boundary may be ascertained by these resulting characteristics. With the advent of high-throughput sequencing, the possibility now exists both to understand and delimit species boundaries based on the analysis of molecular sequences.

The field of DNA taxonomy is based on the use of these sequence data for statistically sound conclusions about species limits, with predictive properties (Wiens, 2007). While in its infancy, a number of delimitation routes are being explored. Distance-based methods such as those of Marshall et al. (2006), on allozyme and restriction site data of the *Sceloporus* lizard species complex, are the most intuitive. The methods used included one based on a cutoff of genetic distance expected at the speciation boundary (Highton, 2000). A second approach known as population aggregation analysis (Davis and Nixon, 1992), is based on the phylogenetic species concept. Here, individuals are clustered based on synapomorphies contained in a population, then these clusters aggregated where distinctions between groups are absent. A related method compared by Marshall et al. (2006) is that of Wiens and Penkrot (2002). This tree based procedure uses concordance/ discordance of a tree from the molecular data with the geographic localities of the individuals. For example, where a focal species from two geographic localities form distinct clades on the tree, this is evidence of lack of gene flow, and a possible cryptic species. When these various methods were used to analyse the lizard species complex, congruence of the results with expected evolutionary groups was partial. Where speciation is recent, it is likely that the lack of character divergences makes species clustering ambiguous. Still, this may simply represent a character sampling issue, which may be overcome using large sequence based datasets.

These methods of species delineation can now be applied on a larger scale as DNA sequence data are becoming available for many species, represented by numerous individuals. Hebert et al. (2004) proposed that species may be delimited to a reasonable degree of accuracy by distance clustering of a single gene. In their analysis of an avian dataset, 2KP distances were calculated for all individuals, and a threshold for a between

species degree of molecular divergence set at an (reasonable) arbitrary value. The authors suggest that where two individuals show greater than 10 times the average intraspecific variation, these are most likely two separate species. They suggest this is accurate in perhaps 96% of cases. However, clustering according a set evolutionary distance does not account for variation in the rate of substitution across subsets of the taxa, which causes overlap in the amount of within and between species divergence estimates. For example Meier et al. (2006) examined Dipteran cox1 sequences and found 6% of cases where cox1 sequences were identical, but obtained from different species. Further, there are many theoretic reasons why the inferred history of mitochondrial sequences will not follow the species history, including the retention of ancestral polymorphisms, introgression, and paralogy (Ballard and Whitlock, 2004). Although the simple inclusion of addition genetic information (particularly, of a nuclear origin) would negate many of these confounding effects (Hebert and Gregory, 2005).

More advanced tree based methods of species delineating from trees inferred from sequence data, include that of Pons et al. (2006). These authors noted the marked increase in branchlength which occurs in the shift from within species branching events to between species events. Therefore, combining a model describing within species variation (coalescent) and between species birth extinction (a Yule process), in a likelihood framework, on a single tree, the 'speciation boundary' may be determined by fitting these as to maximise the likelihood of the data. In cases where multiple loci are used, O'Meara (2010) extends earlier work (e.g. Knowles and Carstens 2007) on species delineation using multiple gene trees. This method exploits the fact that due to gene flow, there would be little congruence of the intraspecific branches. The method performs a search for minimal gene tree conflict only at the interspecific branches.

Such methods are not applicable when the individuals number much greater than 200. With the datasets in use for this thesis, rapid similarity based clustering is one of the few options available. Brock et al. (2009) give one particularly rapid procedure that uses the NCBI program, BlastClust. Molecular distances are used to delimit according to user determined cutoff. But the calculation of distance using pairwise Blast alignment allow for species filtering on raw unaligned sequences, which is advantageous as the computation problem of multiple sequence alignment can be radically reduced by prior

species filtering, or conversely, the computational demand of alignment of a large (non-species filtered) dataset would be substantially greater. For the datasets in use here, this represents the only feasible method currently available. Whilst errors are expected, its use does demonstrate some difficulties in the combination of incongruous datasets (see chapter four).

1.4: Multiple Sequence Alignment

After obtaining DNA sequences for a given locus and filtering unnecessary (within species) data, the molecular phylogenetic analysis proceeds through the main steps of sequence alignment followed by tree inference. The occurrence of indels through evolutionary time necessitates alignment of molecular sequences from different taxa. The rate of indel formation varies greatly both within a gene, and between genes. Functional constraints of the cox1 locus mean indels are rare, meaning straightforward alignment of this gene. For genes that code proteins, accuracy may be increased by translation of the DNA sequences for each taxon, followed by alignment of the protein sequences. The alignment of columns where 20 characters occur instead of 4 characters gives a much lower likelihood of ambiguity. Further, the back translated DNA sequence has more biological meaning, since the reading frame is maintained. In the case of rRNA genes a lack of constraint particularly in the loop regions of the rRNA molecules result in frequent indels. In these cases sequence alignment can be problematic, particularly for divergent taxa. Alignment algorithms are employed which maximise nucleotide homology among taxa, based on various optimality criteria. Most procedures generate gap characters to be inserted into an aligned matrix, while the optimal placement and extent of indels is determined by a balance of the number of indels and mismatches. As numerous sequences are used, alignment in some of the most widely applied methods proceeds in a progressive pairwise manner, typically following a guide tree. Progressively more distantly related pairs (where a member of a pair is either a sequence or a set of already aligned sequences) are aligned to each other according to the tree, until a complete alignment is built. Where indel formation is rapid or included taxa are distantly related, homology assignment may not be possible. The BlastAlign program (Belshaw and Katzourakis, 2005) attempts to address this issue by omitting the non-

matching segments. The program searches for the most representative sequence in the input file, and outputs a Blast-based alignment anchored to this sequence. BlastAlign has been found to perform well in Coleoptera rRNA datasets (Hunt et al., 2007; Hunt and Vogler, 2008), and so is used in the current studies.

1.5: Building Trees From Multiple Loci

Once the alignment of molecular sequences has been established, i.e. the assignment of homologies is fixed, the matrices can now be used to infer evolutionary relationships. Phenetic approaches to tree building use measures of evolutionary distance between sequences to assign relatedness, but these have been surpassed by algorithms that attempt to reconstruct the evolution of states along a tree. As this requires a tree itself, a search is performed where a series of perturbations are made to some preliminary tree in an attempt to find increasing more optimal topologies. Traditional treesearching packages such as Paup (Swofford, 2000) and MrBayes (Ronquist and Huelsenbeck, 2003) run prohibitively slow where taxa number beyond 100-200, and are unlikely to find an optimal tree due to the presence of local optima (Goloboff, 2002). Rapid searching algorithms are being developed to handle the larger datasets commonly used today. For example TNT (Goloboff et al., 2008) avoids local optima by searching subsets (sectors) of the tree, and proceeds beyond local optima it does encounter by tree perturbations (ratchet). In a ML framework, RAxML (Stamatakis et al., 2005) makes substantial savings on computational time by limiting the range of branch rearrangement and likelihood calculations to localised regions of the tree.

The common use of multiple genes has led to two general approaches to their combination, these are termed the supermatrix and the supertree approach (de Queiroz and Gatesy, 2006; Bininda-Emonds, 2004). In the supermatrix method (which is perhaps most commonly used), the different genes for a given taxon are strung together (concatenated), with missing data characters inserted where loci are not available, then the treesearch is run on this single matrix. Alternatively, when using the supertree approach, a separate tree is generated for each of the loci, then these source trees combined. The relative merit of these two methods are subject to discussion (e.g. Gatesy et al., 2002; Gatesy et al., 2004; Malia Jr. et al., 2003). The supertree method may hold

advantages where diverse character sets are in use, or where many existing trees are available (as opposed to the data). However for the current project, the supermatrix approach is more appropriate. The dataset is wholly molecular, with regular development and updating of the analysis methodology. Using a supermatrix approach in these cases avoids character duplication and integration of suboptimal source trees. Where sequence based species delimitation is used, incongruence between loci complicates the generation of supermatrices. This issue is covered in chapter four.

1.6: Determinating Accuracy of Inferred Trees

Phylogenetic simulations evolve sequences along a known tree using established models, then phylogenetic analysis performed on the evolved sequences as they occur at the tips (see chapter five for a simulation study of heterogeneous sequences). In the case of trees inferred from empirical data, the true tree is unknowable, and so determination of phylogenetic accuracy here uses experience of the simulations, in that characteristics of the inferred tree which correlate with distance to the true tree can be used as accuracy indicators. Accuracy indicators can be intrinsic, or they may be derived independently. Perhaps the most commonly used intrinsic estimators is the bootstrap (Felsenstein, 1985). In deciding whether the likelihood of a given topology is significantly better than the likelihood of any other, it is necessary to determine whether the difference in scores falls outside the variance in likelihoods. Variance in the bootstrap is obtained by resampling columns in a MSA replacement, then treesearches (with likelihood calculations) repeated on these resamples. After bootstrapping a value is obtained for each node indicating the frequency the clade is recovered during resampling, low values suggesting that more data (in terms of columns) is required. The repetition of analysis (and therefore, computations) is the major downfall of the test. Kishino and Hasegawa (1989) provide a means of statistically comparing topologies that requires no such repetition, but calculates variance using across-site likelihoods. When such a variance is known, it is common to compare competing phylogenetic hypotheses by placing topological constraints during treesearching and statistically testing the maximised likelihoods (Shimodaira, 2002; Goldman et al., 2000).

The shared evolutionary history of different characters is expected to lead to a given taxon having multiple synapomorphies, across these character sets. Phylogeny is the most likely explanation that independent characters sets share similarities (Sheldon and Bledsoe, 1993), and so congruence is sometimes used as a measure of tree reliability. Congruence may be present within a given dataset (Kitching et al., 1998), or between datasets of different types (Hillis, 1987). As the datasets most frequently used are molecular and morphological, is most often these two character sets upon which congruence is examined. As morphological datasets have historically been used in the definition of taxonomic clades, taxonomic consistency has recently been adopted as a convenient accuracy measure (Hunt and Vogler, 2008). Early indications are that a strong shared evolutionary signal is found between morphotaxa and molecular datasets (Jablonski and Finarelli, 2009), and taxonomic congruence is strongly correlated with other measures of phylogenetic utility (Wilson, 2010). Such taxonomic consistency is typically calculated from the number of taxonomic state changes as mapped onto a phylogenetic tree. A failing of this score is that it does not take into account clade sizes, so for example where a molecular tree finds a taxonomic group of 50 species as polyphyletic with two groups, group sizes of 49/1 or 25/25 would both be scored the same (as both represent a single taxonomic state change). This has prompted the development of a new taxonomic consistency score which differentiates such symmetry/asymmetry (chapter two).

1.7: Factors Influencing the Accuracy of Inferred Trees

The true branching pattern of the tree of life is unknowable, so we can only make inferences based on the available data that estimate the branching order. This requires an in depth knowledge of the way in which molecular data evolves (which in turn we can only assess with the knowledge of the phylogeny). Yet with the application of increasingly realistic models of character evolution we are better equipped to make a reasonable estimate of the evolutionary path (pattern of branching) to explain the patterns on sequence data observed in current day taxa. Also required are indicators of branching ambiguities and branching errors, indicators of the presence of evolutionary phenomena

which could result in these errors, and knowledge of the data structure required to overcome these difficulties and robustly resolve the tree of life.

Early attempts to understand molecular evolution came from the study of protein sequences. The observation of substitutions in the Cytochrome c protein led to the hypothesis that evolutionary change in a molecular sequences is a gradual, constant, clock-like process, such that a given evolutionary time period would result in a degree of site substitutions relative to that time (the molecular clock hypothesis: Zuckerkandl and Pauling, 1965). Not surprisingly, the picture has turned out to be much more complicated than this.

The evolutionary rate in a protein such as Cytochrome c may be quite dissimilar to the rate at proteins elsewhere in the genome (Sharp et al., 1989), and the substitution rate at an individual amino acid is also variable depending on its position in protein (Dayhoff, 1979). As DNA sequences that coded for these proteins became available, equivalent evolutionary patterns were observed. Attempts to statistically describe this pattern of rates found that a gamma distribution provided a reasonable approximation (Rzhetsky and Nei, 1994), with a large number of sites evolving slowly, and a tail of rapidly evolving positions. Further, substitution that lead to change in the identity of a protein were found to happen at a much lower rate than those that did not (Yang et al., 2000).

The complexity of molecular evolution is apparent. These complexities may be prevalent in a given dataset, or they may be absent. Where present, they may be incorporated into a model of sequence evolution, or they may be ignored. The former may be found as statistical tests for such heterogeneities are available (Goldman, 1993; Yang, 1995; Knudsen and Miyamoto, 2001). But in the presence of heterogeneity, what are consequences of attempting phylogenetic reconstruction without accounting for it? In some cases knowledge and modeling of heterogeneity is essential in order to attain a reasonable degree of reconstruction accuracy (Yang, 1996a). For example reconstructing trees without a model of among site rate variation can lead to decreased probability of recovering the correct topology (Tateno et al., 1994), susceptibility to long branch attraction (Lartillot et al., 2007), major underestimation of the length of longer braches (Yang et al., 1994), and high statistical support for incorrect branches (Swofford et al.,

2001). Although it is important to note that the origin of these findings is in simulation studies, and some of the heterogeneities enforced in these simulations may be more extreme than that encountered in nature (Wu et al., 2008).

These among-site rate variation models still assume rate constancy over time, but this assumption also may not hold. Again with the Cytochrome c protein, Fitch (1971b) found that an amino acid which was invariable in some taxa could evolve at a fast rate in others. Recently this has been described more generally, as variation in substitution rate of individual site, over evolutionary time (Philippe and Lopez, 2001). This form of rate variation (heterotachy) has been much less studied than among site variation. It is necessary to describe both its prevalence in commonly used loci, and its effect on phylogenetic inference. I demonstrate the utility of DNA barcodes in the study of molecular evolution, by assessing the impact of heterotachous sites on phylogenetic inference (chapter five).

1.8: Implementation; the Bioinformatic Approach to Phylogenetics and Molecular Evolution

The application of computational techniques to molecular evolution and phylogenetics is known as computational evolutionary biology, whereas the development of the techniques themselves is classed as evolutionary bioinformatics. Evolutionary analyses that would be otherwise intractable are made feasible through these research areas. The techniques used in evolutionary analyses may be custom built within various programming environments, or alternatively, via generalised tools which exist typically as libraries to programming languages (along with the usual stand-alone packages). A substantial part of the work towards this thesis is the use of these bioinformatic approaches in the development of a phylogenetic analysis framework, allowing a test of the suitability of incorporating cox1 (given sampling and heterogeneity issues) into supermatrix phylogenetics.

The working environment for the bioinformatician are programming languages, with high-level (scripting) languages such as Perl and Python very popular. With high-level languages, the basic components from which applications are created are provided, making for rapid development (Ousterhout, 1998). Perl is a scripting language with

particular strengths in text processing, making it ideally suited for processing of molecular sequences (which are represented textually). Additional functionality is also provided via loading libraries, which provide basic algorithms for appropriate for the research area (allowing application development) and interfaces (for pipelining). Most popular languages have libraries available which are specifically geared towards evolutionary applications (Dutheil et al., 2006; Pocock et al., 2000; Paradis et al., 2004; Stajich et al., 2002). With a chosen approach and the necessary tools, it is advantageous to implement the analysis in the chosen framework, such that it may be run with little intervention. In a phylogenetic pipeline, the output for an initial analysis is fed into the next analysis as input. This is achieved by running commands for each step in the analysis, in batch mode, such that subsequent steps are run on completion of the prior command In practice, implementation can be somewhat difficult. For example the format in which DNA sequences appear in a file are not standardised, so most steps require an intermediate subroutine to read an output file and generate an input file in the appropriate format. This is one task for which phylogenetic libraries will provide the tools. This pipelining allows program repetitions far beyond that achieved manually, and opens new avenues of analysis. For example the invocation of an application under a range of input parameters can be automated to determine the sensitivity to these parameters. Also, where these systems are formalised (which consists of building a 'front-end', or user interface), they can be used with minimal effort, by people with little knowledge of the usage of the individual programs. One such service called phylogeny.fr, has been developed and provided as a web service, by Dereeper et al. (2008). The user of phylogeny.fr only has to provide DNA sequences and choose his/her favored analysis options, then all operations necessary to carry out the analysis under those settings, are performed by the pipeline system. This frees up time and effort of the user, which is useful in cases where analysis of the results are of most concern.

Few such pipelines are available, and they are invariably limited in scope. Here I explore new avenues in phylogenetic pipelining, with two features of note. Firstly, I implement recently developed sequence-based species delineation approaches as an option. Also, I automate the incorporation of geographic information at the supermatrix

scale. But the primary use of the (adapted) pipeline is to assess the effect of using DNA barcodes in large-scale phylogenetics.

1.9: Synopsis of Content; Phylogenetic Analysis Utilising Cox1? Problems and Hypotheses

The barcode datasets provide an immense resource for phylogenetics, and could potentially form the backbone of the tree of life. This thesis addresses various aspects of implanting cox1 in large scale phylogenetics. Firstly, does the way in which cox1 is sampled (taxonomically) have an impact on the quality of the phylogenetic inference, i.e. does the rapid rate of change (and resulting high level of homoplasy) still provide phylogenetic information under increasingly dense taxon sampling? Secondly, can we make absolute improvements to existing supermatrices by incorporating cox1, and how many taxa in the combined matrix have to be represented by cox1 sequences are required? Third, I address methodological shortcomings in the formation of supermatrices when incongruent sequence-delineated loci are combined, and finally, how might the resulting datasets be used, and what will they tell us about the process of molecular evolution.

Chapter Two: Taxon sampling density and Braconid wasp (Hymenoptera: Ichneumonoidea) phylogeny using rarefaction and a new measures of taxonomic congruence

2.1: Introduction

Mitochondrial protein-coding genes have played a large role in phylogeny reconstruction, either alone or in concert with other gene fragments. They are easily amplified due to their high copy number and the presence of conserved primer binding sites. On the other hand, the high rates at which they evolve mean they often show saturation of substitutions. Two gene fragments have played a particularly important role in studies of Hymenoptera phylogeny in recent years, the nuclear ribosomal 28S and the mitochondrial cox1. However, both pose their own problems. The former is length variable, and how best to cope with this has led to detailed investigations (Gillespie et al., 2005; Laurenne et al., 2006). Cox1 is far more rapidly evolving, especially its 3rd codon position, and there has been much debate over its utility for inferring phylogenetic history.

Recently, a vast additional source of mitochondrial cox1 sequences has started to become available from Barcode of Life Data Systems (BOLD, Smith et al., 2009) and similar initiatives. The primary aim of these sequencing programmes has been to use the cox1 barcode to identify a species on the basis of genetic distance and sequence clustering. The observed variation also of course reflects evolutionary history, is mostly neutral or nearly so and therefore ought to provide a wealth of phylogenetic information. But first it is necessary to assess the degree and range of phylogenetic signal.

Assessing phylogenetic accuracy is easy with simulation studies but difficult with real data because we do not know the true phylogeny. Assessing success by how well results agree with independently derived hypotheses offers the easiest solution, and as most classifications are based upon morphology, taxonomic hierarchies provide a benchmark for assessing the accuracy of molecular trees. Given a sufficiently large dataset, any biases resulting from the process of taxonomic assignment are expected to be overcome.

Recently, the taxonomic retention index (tRI) has been employed as a measure of the congruence between molecular phylogenies and a prior, usually informal,

morphological classification at family, subfamily, tribe and genus level (Hunt and Vogler, 2008). The tRI treats each taxon (at any hierarchical level) as a pseudocharacter and is calculated as the retention index of that pseudocharacter on the phylogenetic tree. If applied to all taxa at a particular hierarchical level, e.g. genus or subfamily, the ensemble RI provides a measure of fit of the tree to the Linnaean taxonomy on that level. The tRI therefore provides a simple measure of tree quality that varies with the number of clades that an expected taxon is distributed across, with tRI = 1 if an expected group is recovered as monophyletic, and tRI=0 if all its members are dispersed (maximally polyphyletic). Here I introduce a new measure based on the Simpson dominance (SD) index, the taxonomic SD (tSD), which takes into account not just the number of clades that a taxon is split into, but also the distribution of the number of terminals in each of those clades (see below).

Applied to rapidly evolving sequences such as cox1, congruence is likely to decrease with increasing taxonomic and temporal distance between taxa due to phylogenetic signal being lost as substitutions approach saturation. Nevertheless, Kallersjo et al. (1999) showed that the 3rd codon positions of such genes, despite approaching saturation, still contain important phylogenetic signal. Such signal is expected to become useful if taxa are sampled sufficiently densely such that globally homoplasious characters become locally informative. With a simulated phylogeny approach, Purvis and Quicke (1997) showed that it was possible to recover, using parsimony, the correct phylogeny for a very large number of taxa given a sufficient number of nucleotides evolving in a manner not dissimilar to cox1.

The finding that large-scale datasets of short sequences can recover phylogenetic trees with unexpected accuracy (Hillis, 1996) has led to much study in the attempt to understand the process by which the density of taxon sampling leads to improvement (or otherwise) of tree searches (for a review see Heath et al., 2008). Primary among these processes is the dissection of long branches when extra taxa are included. It has been long understood that convergent character states resulting from sustained, unbroken evolution, or increased evolutionary rate, misleads tree building algorithms. Hillis (1996) has shown that the addition of taxa has a dispersive effect on this type of homoplasy.

As phylogenetic inferences may be improved by increased sampling, it is pertinent to ask whether the improvement may be enhanced by the particular method of taxon addition (Hillis, 1998). An intuitively reasonable approach may be the addition of newly acquired sequences to taxonomic (or otherwise delineated) subsets of the dataset under study, as purely random addition may simply add to any current sampling bias. Given existing phylogenetic branching information, a more advantageous approach may involve targeting the additional taxa to long branch regions (Graybeal, 1998), or at least prioritising these regions of the tree. Nonetheless, taxon addition is not expected to have a consistent effect on accuracy across the tree, as improvement appears complicated by context (in terms of local taxonomic and node density).

In the current study I assess the influence of taxon sampling in a newly compiled Ichneumonoidea cox1 dataset. The Ichneumonoidea comprises two huge, cosmopolitan families, the Ichneumonidae and the Braconidae, with 24000 and 17000 described species respectively, distributed among approximately 85 generally recognized subfamilies. The Braconidae in particular has been the subject of a great deal of phylogenetic analysis in recent years (Quicke and Achterberg, 1990; Belshaw et al., 1998, 2000; Dowton et al., 2002; Shi et al., 2005; Pitz et al., 2005), and indeed has become a test case for phylogeny reconstruction in the face of multiple convergent shifts in biology leading to marked conflict between molecular and morphological data sets (Quicke and Belshaw, 1999). Simulations have shown the inconsistency of tree recovery with reduced sampling, but is this substantial in the context of a large and variable phylogenetic space? Sampling within the established Ichneumonoidea taxonomic framework, I find the sampling effect on tree quality is not consistent across taxa, but influenced by the local environment.

2.2: Materials and Method

Our initial data set was compiled including all readily available cox1 barcode fragments of Ichneumonoid wasps. These included all published records available on Genbank, many then unpublished sequences from BOLD, and sequences from the laboratories of various co-authors and other collaborators.

Sequences from ~680 genera and 76 subfamilies were represented, thus including all but four generally recognised subfamilies of Ichneumonidae (viz Adelognathinae, Microleptinae, Nesomesochorinae and Tatogastrinae), and four of Braconidae (Telengainae, Ypsistocerinae, Vaepelinae and Dirrhopinae) and the dubiously placed endemic Australian Trachypetinae and Chilean Apozyginae. Where multiple sequences where available for a Linnean species, or putative species within the BOLD dataset, these were first grouped based on sequence divergence, and one exemplar each was selected based on the most complete sequence (greatest length, fewest ambiguously called bases), or, when multiple equally complete sequences were available, one was arbitrarily selected.

This basic dataset included 4174 Ichneumonoidea cox1 sequences. All sequences were aligned manually with reference to amino acid translation. Alignment was unambiguous though a few species displayed single amino-acid deletions.

Because the data set included several extremely well sampled genera with many very closely related species or putative sibling/cryptic species frequently differing by a small number of bases, these sequences were then further pruned to select a single representative from each cluster of identical or near-identical sequences (sequence variation of less than four bases is considered uninformative). Identical sequences were identified and a single member of each haplotype retained using the following algorithm: Sequences were compared in a pair-wise fashion, with pairs considered identical when sharing all bases at unambiguous sites. Both members of the identical pair are scored for sequence completeness, each additional unambiguously scored base gaining a sequence score of 1. The scoring took the full nucleotide code into account, for example a sequence with a ‘C’ at a given position would be considered 1 base longer than a sequence with an ‘Y’ (i.e. C or T) at the homologous site. The lower scoring sequences are then discarded, or in cases where two sequences are identical at shared sites and are also the same length,

the first sequence in the file is retained. After removal of (near-)identical sequences, a set of 3278 was retained (1890 Braconid and 1388 Ichneumonid sequences, with 671 genera and 76 subfamilies represented). Details of the data set of utilized sequences and their Genbank accession numbers are given in Appendix I.

Tree searches

In order to assess the influence of taxon sampling on tree quality, I performed repeated tree searches on subsets of the dataset, and compared these trees with ones based on the whole data set. I contrast the results of rapid Maximum Parsimony (MP) searches, with those of thorough Maximum Likelihood (ML) searches. For ML, a single ML tree was used as the standard. Whereas for MP, a separate tree search with different seed was used for each comparison with trees generated from a reduced taxa dataset, since such rapid searches produces greater variability in resulting topologies. For MP trees, I used the standard driven TNT search (Goloboff et al., 2008), utilising the constrained and random sectoral search, and tree fusing algorithms (TNT command line option: xmult = replication 5 hits 3 autoconst 1 level 5 multiply css rss fuse 3). ML trees were built using RAxML (v7.2.5) (Stamatakis et al., 2005) using the GTRCAT nucleotide model and 4 categories for rate variation among sites.

Taxonomic congruence (tRI) reaches a rapid plateau during tree searching (Fig. 2.1), so I impose a 1 hour time limit on all TNT tree searches. To speed the RAxML treesearch I use the convergence criterion option (available in version 7.2.5), which quits the search when the Robinson-Foulds distance between results of consecutive cycles drops belong 1%.

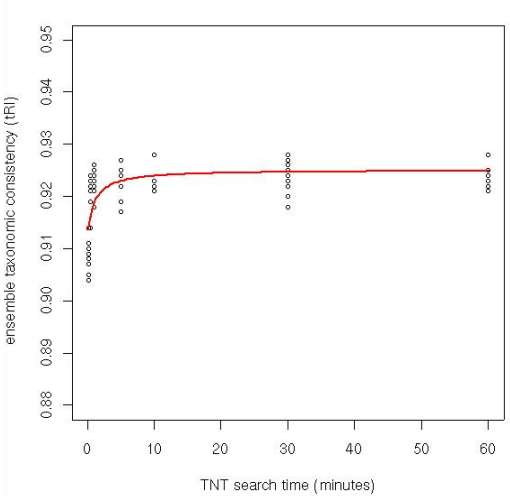


Figure 2.1: Ensemble taxonomic congruence (tRI calculated for all genera) at seven TNT search time limits.

Taxon rarefaction

Three sampling regimes were implemented. In the first, taxa were separately rarefied at the taxonomic level of either genus, subfamily, or family. For the rank under test, the number of sequences belonging to each taxonomic group within the rank were first counted. For taxonomic groups containing more than three sequences, a random number is generated (between 3 and the number of sequences in the taxon), with this number of sequences extracted from the original dataset. 158 of the 671 genera had sufficient sequences (>3) available for the resampling, 54 of the 76 subfamilies were resampled, and at the family level, both the Ichneumonidae and the Braconidae were resampled. Each sequence belonging to the group is given equal sampling probability, and the number of sequences taken were independent across the taxonomic groups of the dataset. After extraction of the data subsets, the individual group samples are compiled to form the rarefied Ichneumonoidea dataset. The second sampling regime used for genus level analysis was rarefaction of a single genus from each subfamily, with complete sampling of its consubfamilials. The choice of genus was random for each subfamily, with the method of sequence subsampling for the chosen genus, performed as above. Finally, in the whole dataset sampling regime I extract a random number of sequences without any regard of their taxonomic affiliation.

.

Taxonomic congruence scores

The tRI was calculated for each taxon using a taxonomic presence/absence matrix. Each terminal (species) in the data set was coded as 0 or 1, depending on whether it belongs to the taxonomic group (e.g. a particular genus or subfamily). The taxonomic matrix for the given group is input into Paup for calculation of the parsimony retention index on a given tree. Sequences that were unidentified to genus level were pruned from the resulting trees so as not to indicate lack of monophyly when the unidentified terminal might actually be a member of the relevant taxon during the calculations of tRI and tSD.

Calculation of the tSD scores started with the 0 or 1 coding and parsimony reconstruction above. Internal branches were then assigned taxonomic status (0 or 1) using the Fitch (1970a) parsimony algorithm. With taxonomic status given for all branches, the number of subgroups that the current taxonomic group had been split into and the numbers of terminals in each of those subgroups were inferred (Fig. 2.2). When multiple switches in a particular taxonomic state were present (i.e. the taxonomic group represented by the state is polyphyletic), the number of sequences in each sub-group were used for calculation of tSD.

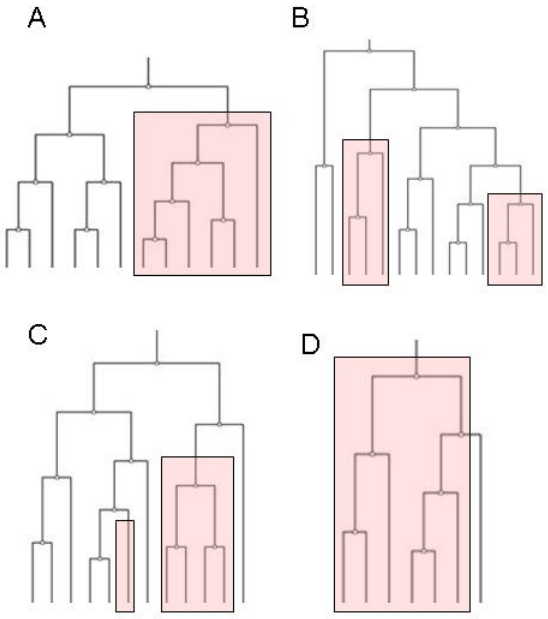


Figure 2.2. Illustrating the behaviour of two methods of scoring taxonomic congruence on four phyletic patterns of a given taxon. Branches belonging to the scored taxon are shown as shaded. The monophyletic pattern in A gives tSD of 1.0 and tRI of 1.0. B shows the taxon as polyphyletic with two equally sized

groups, giving tSD of 0.4 and tRI of 0.8. In C the taxon is also polyphyletic, but of unequally sized groups. Here the taxon scores tSD of 0.67, but the tRI remains 0.8. The paraphyletic group in D (of 5 seqs with 1 invader) gives tSD of 1.0 and tRI of 0.8.

Thus, for example if in a 100 taxon phylogeny a taxon of 11 sequences formed two separate lineages, one of one terminal, the other with 10, the tRI would be 0.9, the same as if they came out in groups of five and six terminals. In contrast, the tSD would be 0.8347 in the first case but 0.504 in the second, thus giving a higher value to the case where most of the terminals formed a monophyletic group. The tSD and tRI respond differently to paraphyly, the former treats a single congruent set of terminals as a single entity irrespective of it being paraphyletic with respect to other groups, whereas tRI effectively responds linearly to the number of instances of paraphyly irrespective of the coherence of the group on the tree.

Branch lengths

I recorded a number of branching parameters for each taxonomic group. These were mean, sum, and variance of branches within the taxon, the subtending branch-length, number of transitions in taxonomic state (i.e. number of times a polyphyletic group is split) and the change in branch-length along the root/tip axis.

Comparison of rarefied and complete trees

The complete trees were used as benchmarks for the maximal expected congruence, against which trees from the reduced datasets could be evaluated. I assess the congruence of resampled and original data set using the method of Poe (1998). Taxa removed in each data rarefaction run were recorded and pruned from the relevant tree built from the whole data set (Figure 2.3)(code in Appendix 1.3). The topologies of the tree from resampling and the whole-data tree (after pruning of all taxa not represented in the resampled tree) were compared based on their tRI and tSD at the genus and subfamily level.

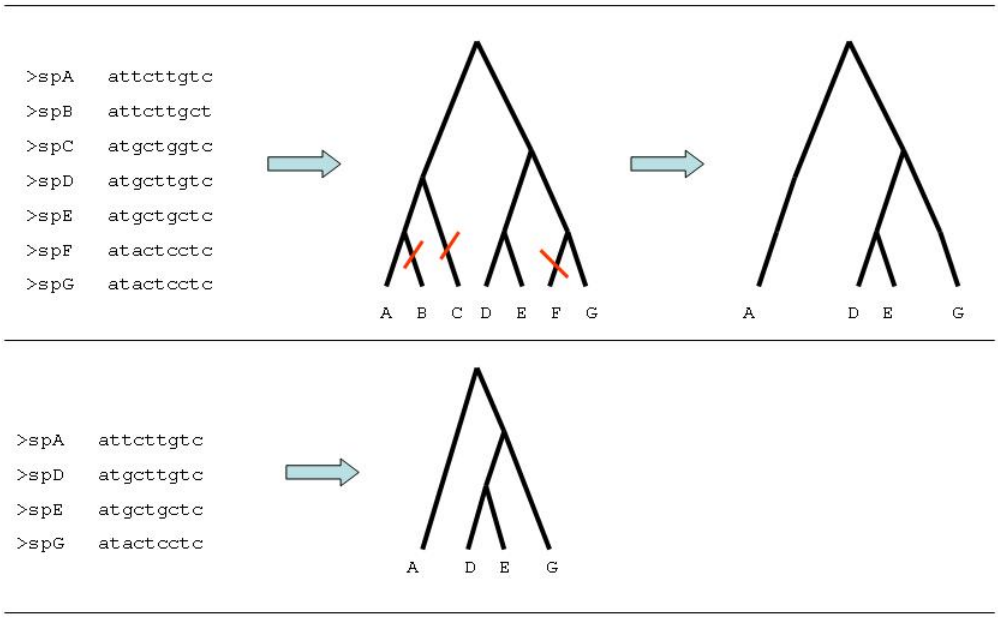


Figure 2.3. Showing how resampled trees are scored by reference to the tree built using the complete dataset, but pruned to contain an identical taxon set. The lower row shows the tree built from the rarefied dataset. The congruence scores from this tree are compared against congruence score from the tree in the top row. This tree is built from the complete alignment (hence using all the contained information), but taxa removed during a given resample, are pruned prior to scoring the complete tree.

2.3 Results

The complete dataset was subjected to treesearches and scored for taxonomic congruence (Table 2.1). At most half of the genera and less than a third of the subfamilies are recovered as monophyletic using these search algorithms. I find differences in monophyly scores between ranks, dependent on the score used. In Braconidae monophyly is greater at the subfamily compared to genus level while the reverse is true in Ichneumonidae. The variation in tRI and tSD is not strongly correlated for these summary values.

	method	proportion taxa recovered as monophyletic		mean tRI		mean tSD	
		subfamily	genus	subfamily	genus	subfamily	genus
Braconidae	ML	0.379	0.430	0.868	0.742	0.792	0.759
	MP	0.276	0.398	0.801	0.702	0.717	0.729
Ichneumonidae	ML	0.278	0.5	0.704	0.713	0.648	0.765
	MP	0.278	0.443	0.719	0.692	0.642	0.747

Table 2.1. Monophyly and taxonomic congruence measures of morphologically defined genera and subfamilies for ML and MP analyses of entire data set.

For each taxonomic level 100 resampled datasets were created, then a separate treesearch was performed on each (Table 2.2). This was repeated with ML and MP search algorithms. To account for the reduction in the number of datapoints for the second sampling regime (sampling of a single genus per subfamily), the number of rarefactions was increased to 600 MP trees and 300 ML trees.

Out of 600 datasets produced for each of the three resampling schemes, a certain proportion was removed because of ambiguous character state optimizations for the taxon names. Results are omitted where there is ambiguity during the calculation of the monophyly score. For example, different ancestral state reconstruction methods (i.e. Acctran versus Deltran) may disagree when assigning branches to taxa. In these cases the count of the number of clades belonging to taxa would be dependent on the method used.

(and so this score is discarded). Cases of ambiguous character optimization are most common in some of the intensely sampled taxa (Ophioninae, Cryptinae and Rogadinae).

While approximately 30% of the subfamilies are recovered as monophyletic in the complete dataset, this proportion drops further when the dataset is reduced. Columns 6 and 7 give the proportion of resampled datasets (for each given subfamily) in which the group is recovered as monophyletic. These correspond well between ML and MP analyses (Pearson's product moment correlation r = 0.944, d.f. = 52, p-value < 2.2e-16). For each subfamily, a mean count of the number of separate clades was calculated, as a measure of the degree of polyphyly, showing that most subfamilies and genera constitute 1.0-5.0 (ML) and 1.0-6.0 (MP) clades in the resulting trees. The taxa found consistently monophyletic irrespective of algorithm and number of sequences sampled in rarefactions were the Charmontiinae, Cremastinae, Hybrizontinae, Lycorininae, Macrocentrinae, Maxfischeriinae and Mesostoinae. Taxa consistently found to be non-monophyletic were the subfamilies Cryptinae, Helconinae and Opiinae, while Doryctinae, Orthocentrinae and Pimplinae break up into the greatest number of separate clades.

A change in the level of congruence with taxonomy (tRI and tSD) in the resampled versus the complete dataset was dependent on the number of sequences left after rarefaction. Splitting the dataset at the median of the number of sequences after rarefaction (three for genus level and six for subfamily level), the proportion of resamples resulting in a different congruence score is significantly reduced on the dataset portion with the greater number of sequences, in all cases. Reporting ML results with MP in parenthesis, at subfamily level, 85% (83%) of resamples in which a small number of sequences are retained do not vary in the degree of congruence with the complete/pruned tree, whereas this proportion drops to 52% (46%) when a larger number of sequences are retained (2-sample test for equality of proportions with continuity correction, chi-squared = 582 (482), P-value < 0.0001). A similar pattern occurs when resampling and scoring at genus level. Poorly sampled genera are unchanged in the taxonomic congruence in 92% (91%) of replicates, whereas a higher sample shows greater variation, 78% (71%) (2-sample test for equality of proportions with continuity correction, chi-squared = 260 (477), P-value < 0.0001).

Subfamily *	# genera included	# of seqs	Number of data points		Percentage of rarefactions in which monophyletic		Mean number of clades recovered	
			ML	MP	ML	MP	ML	MP
Braconidae								
Acampsohelconinae	1	5	78	69	43.6	34.8	1.6	1.7
Agathidinae	4	10	76	71	100	98.5	1.0	1.0
Alysiinae	19	79	86	67	6.1	3.1	1.7	1.9
Aphidiinae	13	56	68	93	7.5	6.3	2.6	2.0
Blacinae	2	12	86	84	8.2	7.4	2.9	2.8
Brachistinae	8	28	65	76	1.8	0	2.1	3.2
Braconinae	40	86	97	98	64.2	50	1.0	1.2
Cardiochilinae	5	13	95	93	52.8	19.2	1.1	2.0
Charmontiinae	1	4	62	64	100	100	1.0	1.0
Cheloninae	7	136	100	96	48	37.5	1.5	2.0
Diospilinae	8	26	92	92	9.4	15.6	3.3	3.1
Doryctinae	57	97	64	59	3.3	3.4	4.4	3.5
Euphorinae	14	84	89	83	3.3	1.4	1.5	1.9
Exothecinae	5	9	84	85	37.5	42.5	1.6	1.6
Gnamptodontinae	3	7	89	90	98.9	94.3	1.0	1.0
Helconinae	10	31	55	53	0	0	1.9	2.6
Homolobinae	2	9	88	91	48.2	39.7	1.5	1.5
Hormiinae	2	9	89	86	6.7	3.6	2.8	2.9
Ichneutinae	3	5	72	70	46.4	25.8	1.0	1.7
Khoikhoinae	2	5	75	78	48.0	62,3	1.6	1.3
Lysiterminae	5	11	89	81	1.4	15.2	2.7	2.3
Macrocentrinae	4	37	93	97	100	100	1.0	1.0
Maxfischeriini †	1	7	83	84	100	100	1.0	1.0
Mesostoinae	7	8	84	82	100	100	1.0	1.0
Meteorideinae	1	3	57	52	100	100	1.0	1.0
Microgastrinae	17	408	98	73	16.7	10	1.2	1.2
Miracinae	1	11	94	90	45.7	34.8	1.5	1.6
Opiinae	6	28	36	50	0	0	1.9	2.8
Orgilinae	3	26	95	85	49.0	6.3	1.0	1.4
Pambolinae	3	6	77	77	36.4	17.8	1.6	2.1
Rhysipolinae	5	6	70	75	2.9	2.7	2.8	2.6
Rhyssalinae	3	5	76	80	81.1	30.1	1.0	1.7
Rogadinae	1	607	22	66	0	1.7	1.0	2.9
Ichneumonidae								
Acaenitinae	4	4	62	62	6.9	11.1	2.2	2.2
Anomaloninae	13	32	96	94	11.5	9.1	2.4	1.6
Banchinae	12	32	87	90	36.8	13.7	1.0	1.1
Brachyscleromatinae	3	4	51	68	86.3	92.5	1.0	1.2
Campopleginae	20	187	50	67	64.3	30.4	1.1	1.0
Cremastinae	6	33	100	97	100	100	1.0	4.9
Cryptinae	97	182	31	31	0	0	2.7	3.6
Ctenopelmatinae	36	73	57	69	1.9	1.5	2.8	2.4
Diplazontinae	12	56	84	77	13.6	13.6	2.6	1.0
Hybrizontinae	2	4	62	68	100	100	1.0	3.9
Ichneumoninae	47	97	86	51	2.6	0	3.8	1.0
Lycorininae	1	5	67	76	100	100	1.0	2.1
Mesochorinae	3	108	96	100	30.4	22.3	2.1	4.1
Metopiinae	11	33	91	87	4.4	2.3	3.5	1.5

Ophioninae	11	150	99	96	66.3	43.5	1.5	5.5
Orthocentrinae	15	112	53	52	1.9	0	3.6	6.0
Pimplinae	20	46	81	58	1.3	1.7	5.0	2.3
Poemeniinae	2	4	85	87	10.1	7.0	1.9	2.0
Rhyssinae	37	3	58	56	3.4	0	2.0	2.4
Tersilochinae	14	23	91	92	4.4	8.7	2.5	5.2
Tryphoninae	1	39	76	69	0	2.9	3.5	2.2

* includes only those represented by more than two sequences
† MS in preparation proposing raising to subfamily status

Table 2.2. Summary of results of subfamily level resampling analyses.

The congruence with the taxonomy is more likely to be lower in the rarefied datasets than in the complete/pruned tree (Table 2.3), indicating an increase in paraphyly and polyphyly with the reduction of taxa. The greater number of cases in which the rarefied tree is of reduced congruence compared to the complete/pruned tree is significant using either MP or ML algorithms, when resampling at either genus or subfamily rank, and for all sampling regimes. Resampling was also performed at family level, but the use of only two taxonomic groups, along with very few cases in which the resampled and complete tree differed in congruence, meant insufficient datapoints for statistical analysis. Still, in all cases of the familial rarefied tree generating a congruence score differing from the complete tree, the rarefied tree gave the lower tSD and tRI.

The graphs in Figure 2.4 plot each taxon (genus or subfamily) in each resampled tree, showing the difference in congruence score between the complete/pruned tree and resampled tree. tSD differs markedly from the tRI at low sequence numbers. Visual inspection of plots of the difference in taxonomic agreement between resampled trees and pruned trees shows a pattern of curves approaching zero with increasing numbers of sequences. These curves are determined by the nature of possible values of tRI and tSD which follow the following equations:

$$tRI = a/(S - 1) + b$$

and

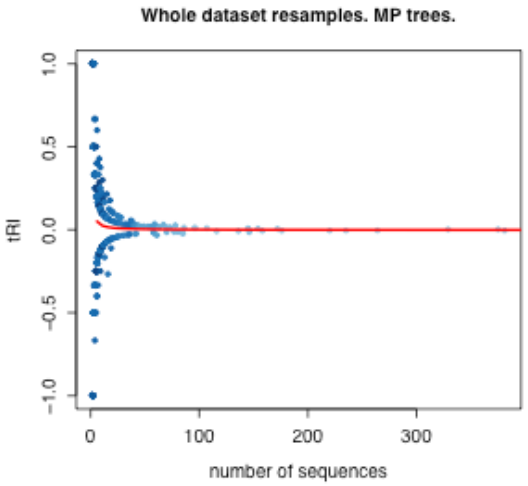
$$tSD = 1 - (S^2 + a^2/2 - aS)/S^2$$

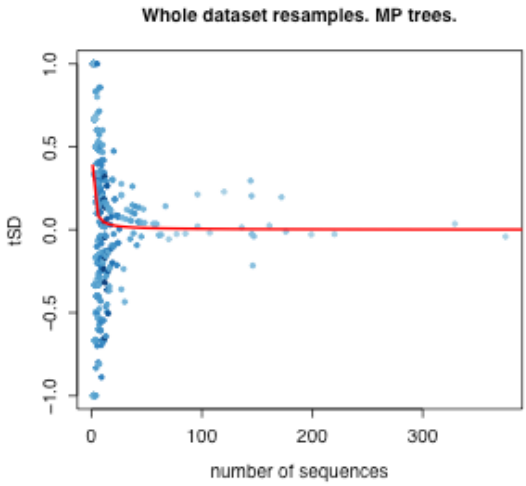
where S is the numbers of sequences and a and b are the fitted parameters. These equations were used to create the lines of best fit in R using the equation-fitting function *nls* with appropriate seed-values for fit parameters a and b.

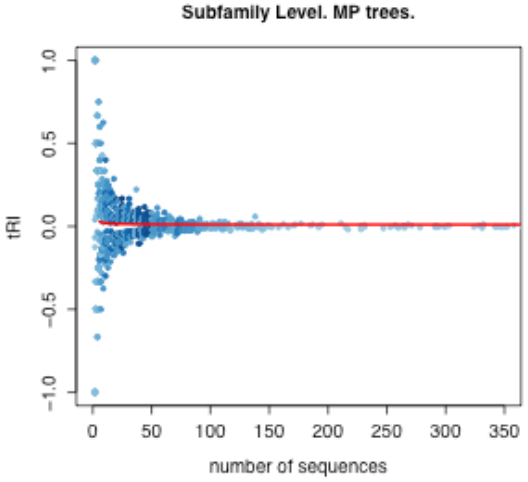
search method	taxonomic rank	p-values tRI	tSD
MP	whole dataset	1.31E-06	0.00028
	subfamily	3.83E-06	0.0093
	genus	5.61E-05	4.85E-07
ML	whole dataset	<2.2E-16	1.42E-10
	subfamily	0.0003	0.0108
	genus	7.56E-10	4.19E-05

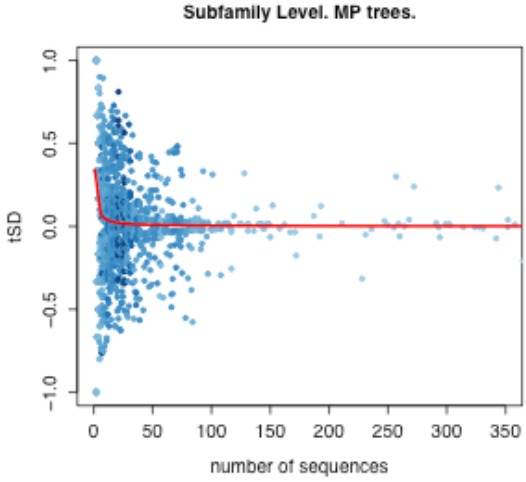
Table 2.3. Chi square probabilities for number of clades for which the pruned tree (based on analysis of whole data set) is more congruent than the equivalent rarefied trees. Values were calculated based on resamples that included more than three taxa.

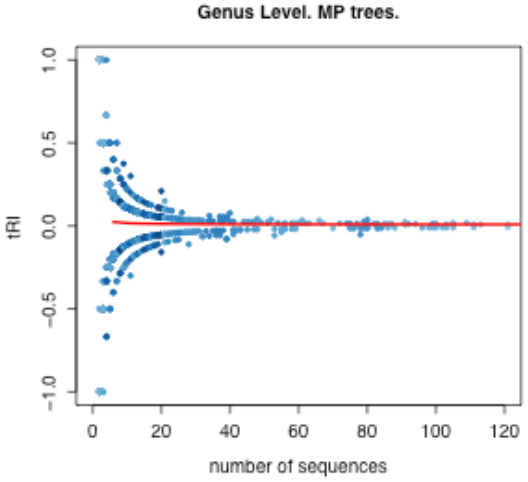
I found the degree of monophyly to be quite robust to changes in the level of sampling. In the majority of cases the congruence scores do not differ between the resampled tree and the pruned tree, indicating a specific sample of taxa form similar placements with or without the inclusion of the complete taxon set. Further, a subset of taxa are consistently monophyletic, in spite of the sample of the taxon itself, and the combination of taxa elsewhere in the dataset. The pie charts in Figure 2.5 give the proportion of taxa at the given rank which are found either i) consistently monophyletic during resampling ii) mostly (95%) monophyletic, or iii) not consistently monophyletic. In general there is a high correspondence between these scores.

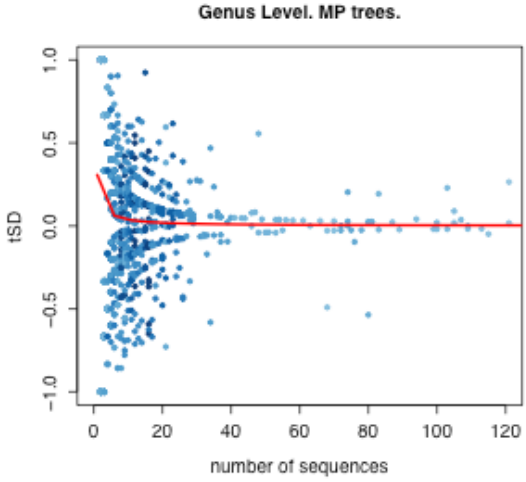

Whole dataset resamples. MP trees.
tRI
number of sequences


Whole dataset resamples. MP trees.
tSD
number of sequences


Subfamily Level. MP trees.
tRI
number of sequences


Subfamily Level. MP trees.
tSD
number of sequences


Genus Level. MP trees.
tRI
number of sequences


Genus Level. MP trees.
tSD
number of sequences

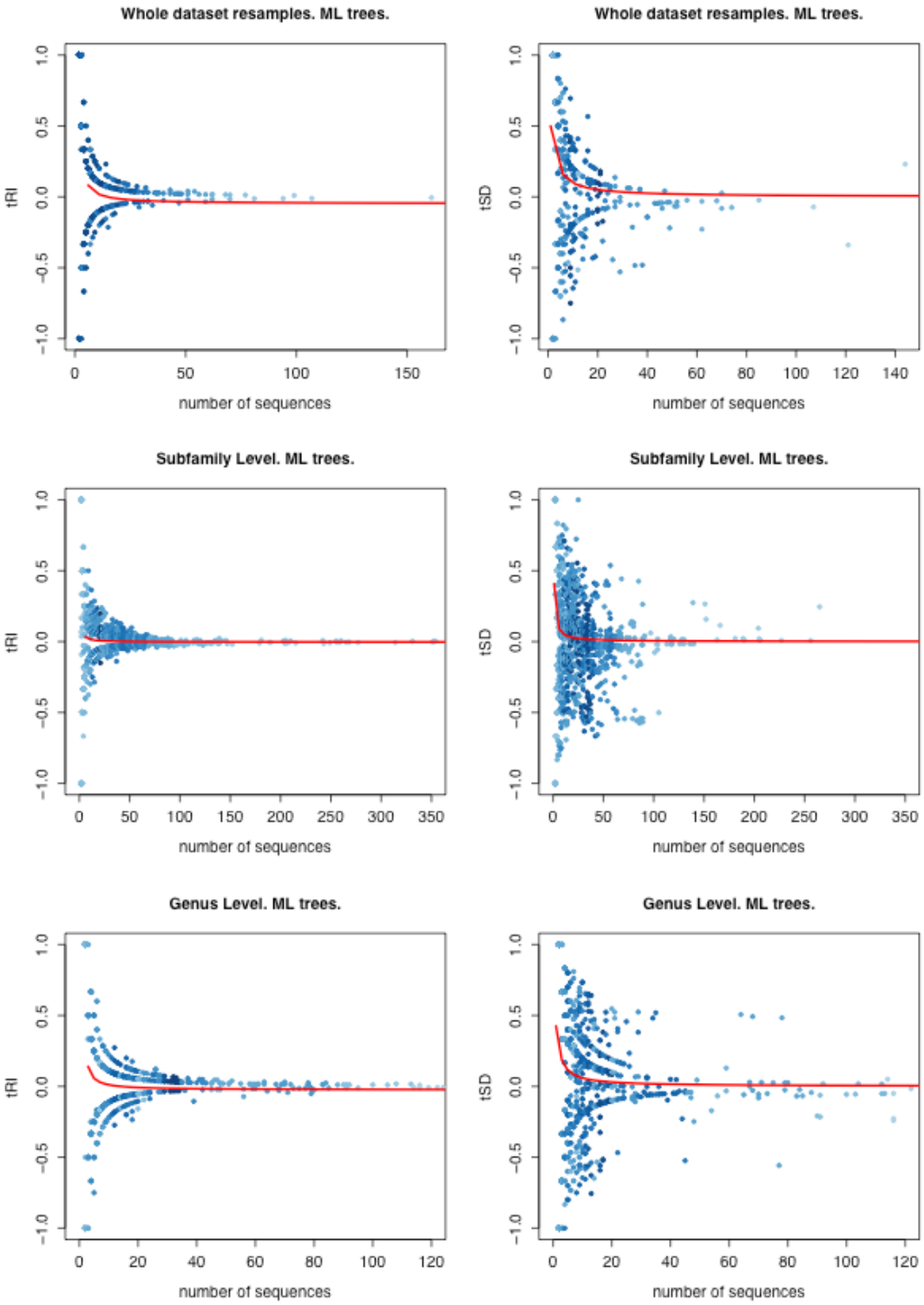


Figure 2.4. Plots of differences in tRI (left column) and tSD (right column) measures of phylogenetic congruence between resampled trees and pruned trees versus the number of sequences retained after rarefaction based on MP analyses (upper six), and ML (lower six), with three different levels of taxonomic constraint on taxon representation.

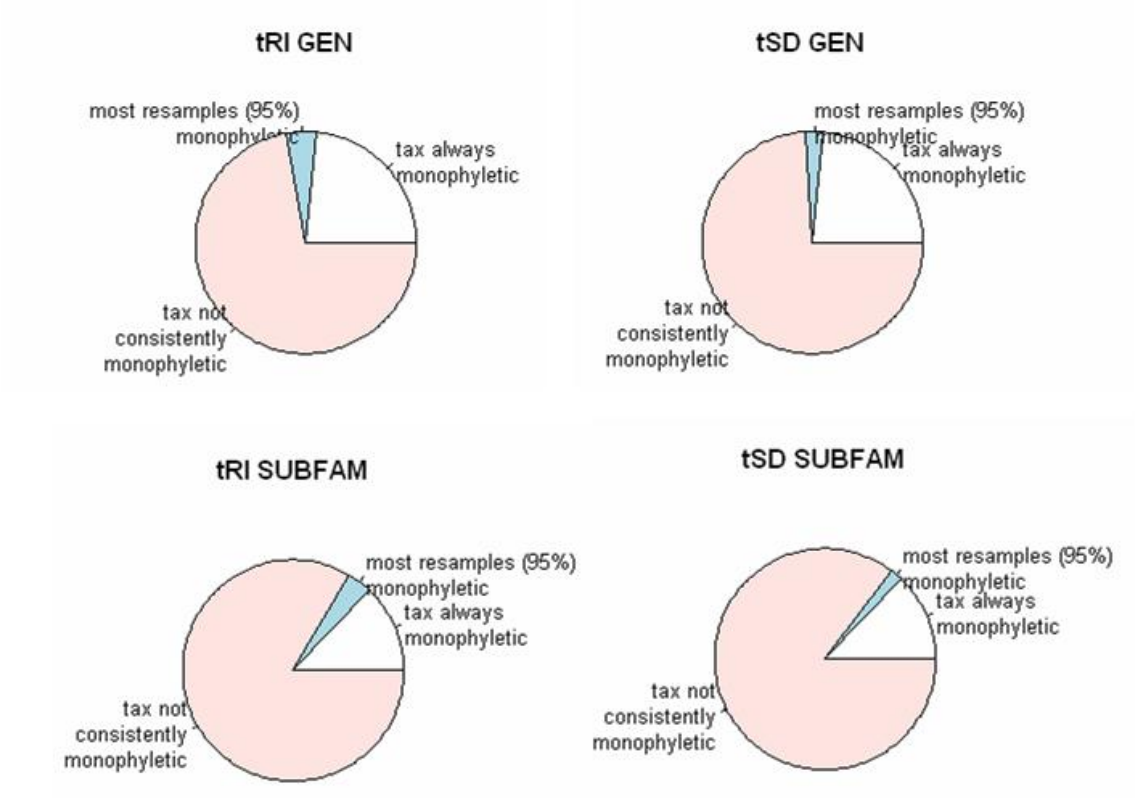


Figure 2.5: Pie charts proportioning the taxonomic groups according to propensity to remain monophyletic during rarefaction.

While the level of sample completion of the taxon itself impacts the accuracy of the resulting tree, the influence of sampling in the local environment is also of interest. Generic sampling within subfamilies differs substantially across the Ichneumonoidea. Here I find this local sampling density influences the degree and variability in congruence of individual genera within. Genera with fewer consubfamilials (a term I use to refer to the number of other genera represented in a taxon's subfamily) are i) more likely to be monophyletic, and ii) show a stronger tendency to increase in congruence upon increase in sample size. Firstly, using the uncorrected congruence scores, I find a greater degree of congruence in regions of the tree which have fewer genera represented. Both search algorithms show a significant negative correlation in uncorrected congruence score (tRI) against the number of consubfamilials, indicating an increase in congruence (monophyly) where fewer consubfamilials are represented (Pearson's product-moment correlation, p-value < 0.0001 for ML and MP trees).

I hypothesised that taxonomic groups within densely sampled regions of the tree would be more prone to change in the degree of congruence than groups in weakly sampled regions, but the evidence was not substantial. Each resample was classed as either robust or non-robust, depending on whether the congruence score varied between the pruned and resampled trees. Each datapoint (now a 0 or 1) was paired with its value for the number of consubfamilials, then the scores divided at the median. I find significant decrease in the number of robust resamples in the dataset half with the higher number of consubfamilials, in the case of the ML trees, but a non-significant increase in the MP test. While this indicates there is sometimes a tendency for poorly sampled regions to remain relatively robust under varying sample size, I find that in cases where the degree of congruence *does* change, there is a strong tendency for improved congruence. Resamples were classed according to whether the reduced dataset generated a tree with less taxonomic congruence than the pruned tree (as is generally the case, indicated by earlier results). In all cases I found an increase in the proportion of this class in the half of the dataset with the lesser number of consubfamilials, but these were only statistically significant in the subfamily and genus level rarefactions.

The current study design differs from most sample size analyses by the simultaneous resampling and scoring of numerous taxa, in a large-scale phylogenetic context. I have shown that phylogenies may still be improved by increasing sample, even within this environment. To determine the influence of this simultaneous resampling, I implemented a second sampling regime whereby the rarefaction and congruence scoring was limited to a single genus from each subfamily, with complete sampling of the remaining consubfamilials. Under this sampling regime the proportion of replicates in which there was no change in the degree of congruence between the resampled tree and the pruned tree was higher (ML: 0.902, MP:0.844) than when all genera are resampled and scored (ML:0.848 MP:0.797). Of the resamples in which the congruence score differs, the proportion in which the complete tree gains a greater score than the reduced tree is also greater than previous (ML:0.617, MP:0.596 under the current regime, as opposed to ML:0.566, MP:0.537 when all genera a resampled). Several factors may account from these changing proportions. With the sampling of a single genus from each subfamily, there is a reduced representation of genera from (generically) densely sampled

subfamilies, which have been shown previously to differ in response to resampling. The down-weighting of these genera can be accounted for by reanalysis of the standard sampling regime, but for each resample, scoring one randomly chosen genus per subfamily. When normalised in this way, the number of resamples unchanged in the congruence score (ML:0.858 MP:0.798) is almost identical to the non-normalised dataset. However, the proportion of cases where the resampled tree has lower congruence than the complete/pruned tree is raised to a value (ML:0.589 MP:0.564) not significantly different to that of the second sampling regime (ML:0.617 MP:0.596) (2-sample test of equality of proportions with continuity correction. χ-squared = 1.4286, d.f. = 1/ p-value = 0.232. When not normalised, χ-squared = 10.3211. d.f. = 1, p-value = 0.001315 ML: χ-squared = 1.041, d.f. = 1 p-value = 0.747. When not normalised, χ-squared = 3.166 d.f. = 1, p-value = 0.0752). This indicates that the concurrent resampling of consubfamilials (or the variability induced by it) creates greater variation in congruence (the proportion of robust resamples reduces from 0.844 with a single genus per subfamily to 0.798, x-squared = 30.3056. d.f. = 1. p-value = 3.69e-08. 2-sample test as previous. ML:0.902 to 0.858 χ-squared=9.1337, d.f.=1, p-value=0.00251) but does not substantially affect the ability of the complete sample to better recover monophyly.

2.4: Discussion

The quality of an inference of phylogeny is dependent and influenced by many factors. These factors include methods used in the analysis of the dataset (choice of alignment parameters or search algorithm) and characteristics of the dataset itself (e.g. heterogeneous patterns of sequence evolution). The dataset used is almost always derived from a taxonomic subset of the actual diversity of the group under study, and as such, represents an intermediate step in the attempt to elucidate the true phylogenetic history. To advance the inference, one may attempt to improve the analysis of the current dataset (perhaps by improving the fit of the evolutionary model) or alter the dataset in such a way as to create one which may be more amenable to the phylogenetic methods under use. Studies suggest that the inclusion of a greater taxonomic sample reduces susceptibility to biasing characteristics during analysis, and increases the similarity of the resulting trees to the true phylogeny. Here I find that the increase in quality is still significant in the context of a 2000+ taxon, variable tree space. However, in the majority of cases where taxa are added, there is no change in the degree of congruence. Although It cannot be inferred there has been no improvement, the presence of topological variation has not altered the level of polyphyly or paraphyly. Only in a minority of cases, is the degree of polyphyly/ paraphyly changing. On average there is a tendency for the phylogeny to improve, but the degree of scatter seen in the graphs (fig 2.4) indicate that for any particular case of taxon addition, an improvement in quality is not assured. For example (when scoring tRI on ML trees) where adding the complete taxon set to a genus leads to change in congruence, in 57% of cases it will improve, but 43% of cases will result in reduced quality. So where taxa are added to a limited number of taxonomic groups, there is little guarantee that increasing sample would lead to improvement. Only when dealing with many groups, would the trend of increasing quality be apparent. Secondly, the absolute number of taxa in the group will impact the response when further taxa are added. There is no substantial change in tRI when adding taxa to a group which already contains ~20 sequences (Fig. 2.4). However, this is not necessarily the case when scoring tSD, with changes of 0.5 observed when even ~ 80 sequences are present in the reduced dataset. Conversely, a considerable difference in congruence scores is often observed on adding sequences to a taxonomic group with few existing sequences, with a strong

inclination in tSD improvement. The greater improvement of tSD compared with tRI suggests the correction of polyphyletic taxa split over similarly-sized subclades. So the evidence suggests a greater benefit to taxon addition in cases where existing sampling is poor.

The evolutionary divergence of the added taxa also substantially impacts the resulting change in quality. Previous authors have suggested bisection of long branches as a likely candidate for which increased taxon sampling improves phylogenetic accuracy. In the current study I find that cases with a marked increase in congruence post-taxon addition are those in which the total branch length has been reduced. So when taxon set placement is optimal (maximally congruent), this is reflected by similarly optimal (or relatively reduced) branching parameters. It follows that the reduction in quality when only a subset of taxa are present, is associated with greater overall branch length. It is uncertain whether this is an overall reduction in branch length, or bisection of a small number of long branches. Still, that such a substantial correlation is seen when measured against the taxonomic framework is intriguing, the branch length reduction following taxonomic monophyly corroborates the placement of these taxa as valid evolutionary groupings.

In general, the implementation of taxon addition is based largely on intuitive decisions made using prior phylogenetic knowledge, including known problematic regions. Simulation studies are proving useful in detailing the mode in which the precise placement of new taxa leads to phylogeny improvement. But given cases where prior phylogenetic understanding is poor, the question remains as to how would one best proceed in taxon addition? Given a poorly known dataset, I have shown the taxonomic sampling alone predicts response to different taxon sampling strategies. There is the choice of taxon addition from regionally sparse or dense regions. The phylogenetic quality increases seen with taxon addition are not consistent across groups. Adding to a subfamily with few represented genera will be less likely to lead to a change in quality, but on the other hand any improvement will be more marked. In this case the results do not suggest an obvious course of action. But taking an alternative perspective, the results are informative on the origins of monophyly. Given an incomplete sample, what are the characteristics of the taxonomic subsets which are most likely to be recovered as

monophyletic? Here it is seen that a subfamily with few genera sampled will have a higher likelihood of showing generic monophyly, and a greater improvement in generic monophyly when sample size is increased. Intuitively, this may be expected as genetic distances would be greater between the sparsely sampled genera, and so perhaps the sparse subfamily represents a simpler phylogenetic problem, in which corrupting effects of a low sample are more easily rectified. Further, the statistical likelihood of monophyly can only increase as consubfamilial sampling is reduced. So, the increased congruence may simply be reflecting the more likely cases of true monophyly.

The above raises more questions: If generic monophyly improves more in sparsely represented than in densely represented subfamilies after taxon addition, must we therefore look for a general correlation whereby increase in distance to the nearest consubfamilial clade allows greater phylogenetic improvement with increased sample. What are the characteristics created by this distance that gives an amenable environment? And, finally of course, what are the phylogenetic errors being rectified under this environment?

The results presented here predict monophyly patterns based on the taxonomic structuring of the dataset. One characteristic of retrieved barcode data is inconsistent sampling across taxa. Sequencing studies are often comprehensive for specific genera or subfamilies, with sparse representation for intermediate groups. Simple reference to the taxonomic labels attached to sequences would be enough to give indications of the likelihood of congruence in dataset subsection, with perhaps unreliable monophylys apparent where consubfamilial representation is poor. Of course it is not certain to what extent these findings are dataset specific. Nonetheless, modern bioinformatics approaches will allow similar analyses on multiple large scale extracted datasets, complementary to the curated approach used in the current study.

The use of the cox1 barcode locus for this study make the results especially pertinent in the light of its adoption in molecular taxonomy. The large scale sequencing efforts for this locus mean it will soon dominate sequence databases, so there is high priority in determination of its (particularly, phylogenetic) utility. Recent work has described the utility and signal in insect COX1 sequences at varying taxonomic rank (Wilson 2010). The author found a reduction in signal in the deeper (subfamily to

superfamily) taxonomic nodes. In the current study I account for this variable signal when assessing the response to change in sample size, by reference to the complete/pruned trees. When this is done there is a surprising level of consistency, with 68% of subfamily and 85% of genus level resamples showing no change in congruence to the complete dataset. For the most part a given taxon combination results in a specific degree of congruence, irrespective of the combination elsewhere or the presence of the full taxon set. This consistency is reduced at the deeper nodes (although this might be expected due to higher sample numbers dealt with at this level). Thus, the signal provided by cox1 appears consistent, but of course whether this signal is unbiased and accurate, is a separate matter.

Using recovery of taxonomic grouping as an indicator of accuracy, roughly one third of subfamilies and just under a half of genera are recovered as monophyletic from analysis of the full data set in the ML trees. One limitation of the current study is the relative incompleteness of taxonomic work on the Ichneumonidae (Quicke et al., 2009), with several traditional subfamilies appearing as either polyphyletic or paraphyletic based on combined morphological and 28S rDNA sequence data analyses. Nevertheless, of the remaining well-supported groups, the cox1 trees still fail to recover many as monophyletic and of course few would claim a single locus in isolation held sufficient phylogenetic information. But it may be assumed slow evolving loci would be necessary to compliment cox1, and I have demonstrated that deep level nodes may still be improved by increasing representation of this locus.

It is important to note that under the methods used here, any change in quality is scored relative to the complete dataset/tree, not the true tree. As such, any case of substantial improvement upon complete sampling may still represent an unsatisfactory level of congruence. The absolute (uncorrected) congruence scores given in table 2.1 represent the maximum attainable for the current dataset. Improvement beyond this presumably requiring additional locus sampling. The behaviour of cox1 in isolation does not necessarily predict behaviour after increased sampling in a multi-locus dataset. Consider a hypothetical existing node perfectly resolved using an existing dataset. Given the addition of a sufficient cox1 sample with the attainable scores found here, would this

be a hindrance to approaching the true tree? In the next chapter I extend the work to assess increased cox1 sampling in multi-gene datasets.

Chapter Three: Improving phylogenetic accuracy of existing supermatrices by concatenation of COX1 data.

3.1: Introduction

The use of single loci in phylogenetic analysis is all but extinct. The theoretic basis for bias resulting from such insufficient character sample is strong, such that four, eight, and more loci are commonly employed, even when the taxon sample under study is large (McMahon and Sanderson, 2006; Goloboff et al., 2009). But sequence databases are somewhat polarized in locus sampling, with a small number of very widely sampled genes, and a large number of genes represented by very few species (Sanderson et al., 2003). This is reflected in studies of taxon/character sampling, either focused on required number of loci in a phylogenomic context, or absolute character addition in dense taxon trees. There is ongoing discussion of the relative benefits of increasing sampled taxa or number of characters, to phylogenetic accuracy. This is a simplification in the context of supermatrices used widely today, as the increase in characters typically means the use of additional discrete loci. Loci have unique evolutionary properties, thus attempting to improve trees by inclusion of an extra locus is not simply determined by the number of extra characters, but many factors relating to the identity of the character set. The heterogeneous nature of sequence evolution has been extensively covered (Yang, 1996a; Yang, 1996b; Hasegawa et al., 1993; Lopez, 2002; Tamura, 1992), with evolutionary rate, character frequencies, relative among site rate patterns and indel frequencies showing variation across the genome. Such evolutionary patterns are known to affect the phylogenetic utility, often because the optimality algorithm is ill-equipped to deal with them (Felsenstein, 1978). Methods to accommodate heterogeneous evolution are ongoing (Pupko et al., 2002; Steel, 2005), but depending on the type of dataset, such independent modeling of all loci may not be feasible. Still, any biasing characteristics are present at varying degrees, and so the likelihood of tree bias would amongst other things, depend upon the genes from which a phylogeny is inferred. Where mined data is used, the choice of loci used is not under the control of the researcher, and so the use of suboptimal loci may be unavoidable. If commonly used loci are indeed suboptimal, with evidence of

biasing characteristics, what are the consequences of incorporating them into large scale datasets?

Locus coverage for any given taxonomic group is invariably an ongoing project. Major published studies will typically provide the framework of the dataset, with the publication of new datasets giving new possibilities in terms of improving this framework. But the number of genes for which sampling is taxonomically dense, is a small proportion of the genome. As such, genome-scale descriptions of phylogenetic information content are somewhat unhelpful for the practicing phylogeneticist, when what is required is detailed description of the genes at hand. Knowledge of these 'phylogenetic loci' is at a premium, rather than locus sampling on the whole. Assuming a reasonable existing dataset, the researcher would like to know whether the addition of locus X, with taxon sample of Y, would lead to a better or worse phylogeny, which may not necessarily be determined post-inclusion (Philiips et al., 2004). The locus would have its own unique evolutionary characteristics, characteristics appropriate for resolving nodes under a certain evolutionary range, and perhaps liable to bias under other conditions. For example the cox1 locus (which has been extensively studied) is known to be fast evolving with optimal character variation at species and genus level, but showing evidence of saturation (Roe and Sperling, 2007). The utility and susceptibility to bias of such a gene would primarily be sensitive to the taxonomic sample size Y (Hillis, 1996). Where the number of sequences used to infer phylogeny is increased, the tree tends towards increased branching similarity with the true tree, most likely because long branches (which are known to erroneously attach to other long branches) are broken (Poe, 2003). Assuming the reason for locus inclusion is increased phylogenetic accuracy, it is desirable to incorporate loci where sampling is dense. Therefore, were behaviors of phylogenetic loci well documented, locus suitability may be predictable based simply on the identity and taxonomic sample size/distribution of each.

Affecting any putative change in quality upon inclusion of extra loci would be not only the heterogeneous characteristics and absolute sample size as covered above, but additionally, the structure of the sample in a taxonomic sense, and in relation to the existing matrix. Firstly of course, 10 sequences from a given genus for locusA may have no species overlap with 10 sequences for locusB sequenced for the same genus. Unless

generic hybrids were created, this taxonomic structure would give no additional phylogenetic information for the existing nodes. A less extreme example where partial overlap were present, is less clear, and the course taken is as much under subjective requirements. It may be assumed that the introduction of new nodes is desirable given the ultimate goal of the species level tree of life. However missing data is introduced, the effect of which is unpredictable (Wiens and Reeder, 1995; Wiens, 2003; McMahon and Sanderson, 2006). In some cases any benefit from increased number of characters may be outweighed by factors such as computation burden, skewed taxon representation and difficulty in reconstructing state changes over large evolutionary distances. So the degree of overlap with an existing framework, and not the absolute sample structure, would be a further determinate of the utility of each new locus.

The cox1 locus is particularly relevant to the question of locus addition. Ongoing barcoding efforts mean including it in existing phylogenetic matrices will become an option available for many taxa. There are some researchers who question the phylogenetic utility of cox1 (e.g. Hajibabaei, 2006). The sequence variation is thought appropriate for the boundary between populations and species, and so state changes may become saturated beyond genus level nodes. Were this the case, then inclusion of cox1 would be expected to provide little increase in phylogenetic signal apart from only the tip level nodes. But could any shortcomings be overcome were sampling substantial? In the current chapter I ask how may an existing supermatrix be improved (if at all), by the addition of cox1.

3.2: Methods

HIP-DB pipeline development

To aid rapid and repeated phylogenetic analysis, we develop a phylogenetic pipeline in the Perl environment. HIP-DB is summarised in Figure 3.1.

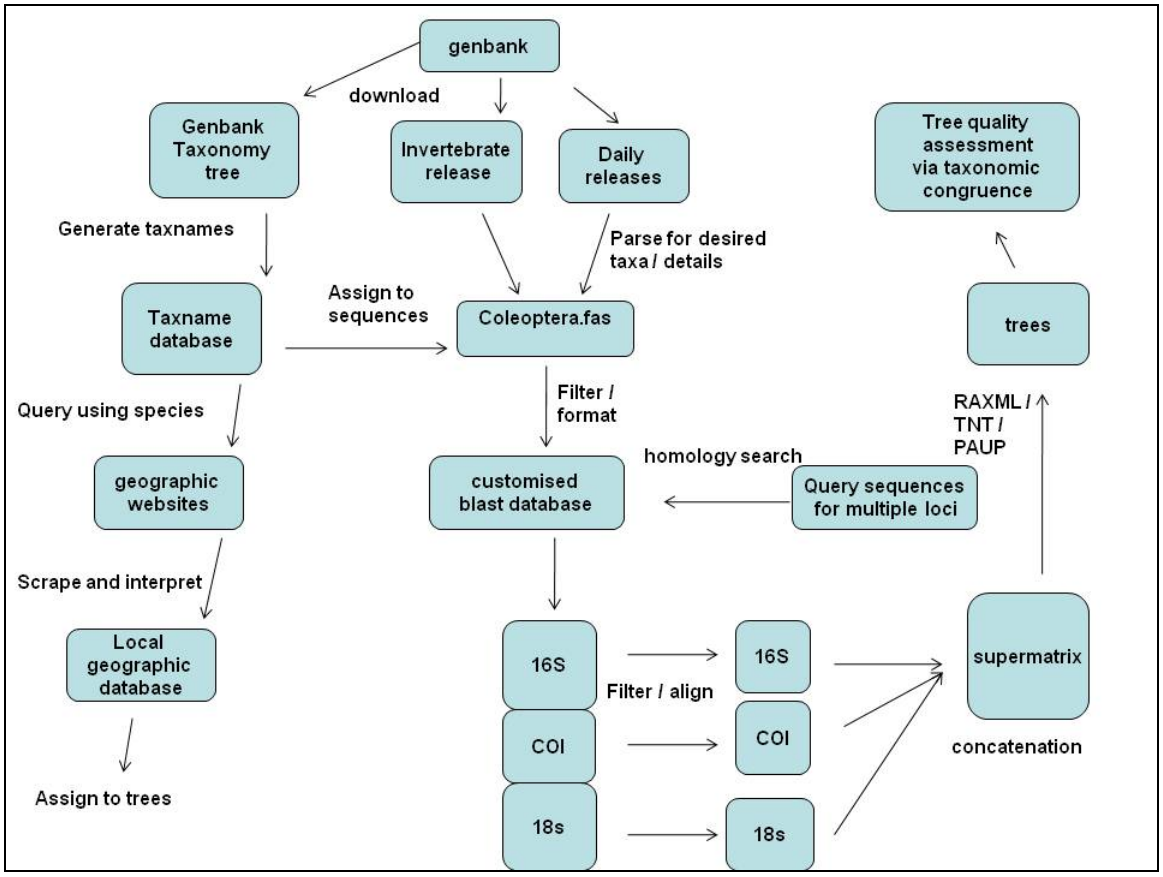


Figure 3.1: The HIP-DB system (Holometabolous Insect Phylogenetic DataBase). The invertebrate release and subsequent daily NT releases are automatically downloaded using the 'wget' linux approach. These files are parsed for user specified taxa (e.g. all Coleoptera sequences). Also downloaded is the Genbank taxonomy database, from which taxstrings are generated (see chapter four). All species names for the specified taxa are submitted to geographic databases to obtain geographic ranges, which are later applied to the trees (where appropriate). Tax strings are tagged onto sequence files and a blastable database formed. User specified loci are obtained by blast searching, then these files filtered to leave one sequence per species. This is achieved either by referring to the species name, or by a sequence-based procedure (chapter four). The sequences are aligned using the user specified method, then the different loci concatenated. Trees are inferred from the supermatrix using scalable software Paup (NJ), TNT (MP), or RAxML (ML). The trees are then scored for taxonomic consistency.

The automation of software invocation, file processing, etc in a scripting environment allows repeated analysis of large datasets. A number of such pipelines have been

developed (Jones and Blaxter 2006; McMahon and Sanderson 2006; Sanderson et al., 2008; Golobof et al., 2009; Thomson and Shaffer, 2010), but are specialised towards specific tasks (usually the authors preferred method of phylogenetic analysis). HIP-DB gives the user a number of options, at points in the analysis which are of particular importance, and also introduces new approaches. For example the method and parameters used in multiple sequence alignment are a major determinant in the resulting topology (Wong, 2008; Borrow, 2010). Thus HIP-DB give the user ~12 alignment options, including: Muscle (Edgar, 2004), HMMalign (from the HMMER package: http://hmmer.wustl.edu), Mafft (Katoh et al., 2002), TransAlign (Bininda-Emonds, 2005), POA (Lee et al., 2002), and a procedure used by Hunt et al. (2007), whereby the sequences are first aligned with BlastAlign, upon which a TNT tree is generated, then this is used as a guide tree for the final alignment in ClustalW (Thompson et al., 1994). HIP-DB introduces some unique features. Species filtering (removal of multiple sequences for each species) may be performed using either the taxonomic names, or by sequenced delineated clusters (see chapter four). Also unique for HIP-DB is the geographic assignment step (example in Figure 3.2). The automated assignment of geographic ranges for taxa in the database requires interfacing with online geographic databases (http://zipcodezoo.com/; www.unep-wcmc.org/isdb/Taxonomy/; www.catalogueoflife.org; http://www.faunaeur.org/), the retrieved ranges are then appended to terminal labels in the phylogeny (code in Appendix 1.6).

Various parts of the pipeline were used according to the task at hand. For example multiple locus searching and concatenation steps were bypassed where analyses were performed on a single gene. Analyses in chapters three and four used much of the pipeline, with additional code appropriate for the analyses.

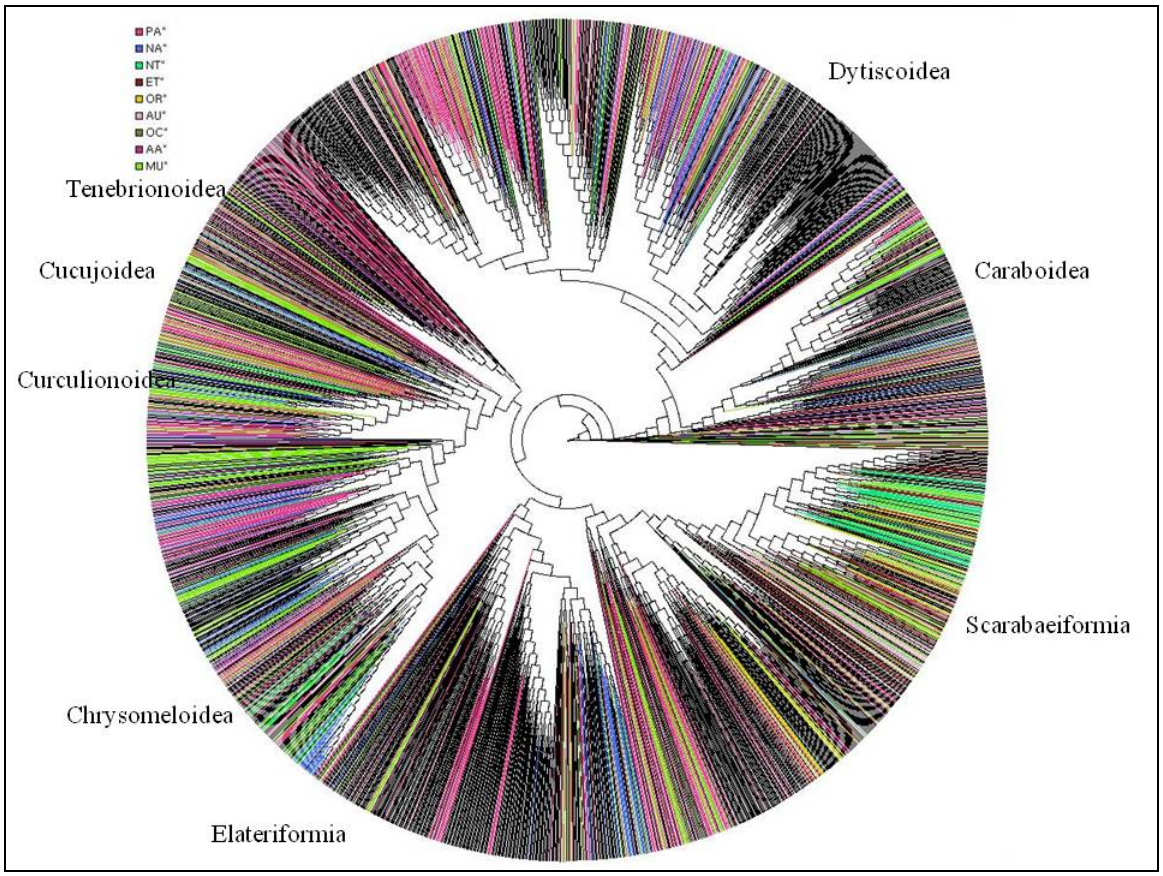


Figure 3.2: An example Coleoptera tree showing assignment of geographic zones to taxa in a phylogeny. Branches are coloured to represent geographic assignments according to submissions to online geographic databases. Zones included are PA: palaearctic, NA:nearctic, NT:neotropical, ET:Ethiopian, OR:oriental, AU:Australasia, OC:Oceanic AA:Antarctica, MU:Multiple, Black branches unassigned. Major clades of the Coleoptera are indicated.

Dataset creation (code in Appendix 1.2)

The HIP-DB pipeline was used aid the assessment of the impact of adding cox1 to an existing Scarabaeoidea rRNA dataset. All Scarabaeoidea sequences were downloaded from the Genbank database, then formatdb used to create a local Blast database. The database was searched for cox1 (3 prime), 16s, 18s and 28s sequences, using multiple queries from a diverse range of Coleoptera families. A Blast search was performed using the e-value cutoff of -e 1e-5, then blast output files parsed and the hits extracted (some BioPerl code was used here, see Stajich *et al*.2002). For each hit we extract the database id, the start and end position (in terms of the database sequence) and the strand. As multiple queries are used, multiple hits are made to the same database sequence, but as these are for different queries, the start and end positions do not always correspond. Thus

the left-most and right-most positions are found for each database sequence found to be homologous to the queries. After parsing the blast output file, each hit is extracted from the database using the NCBI program, fastacmd. Sequences were extracted and trimmed to the start and end sites, and the opposite strand inferred where necessary. This procedure was repeated for each locus, giving four corresponding files. The files were scanned and where duplicate sequences for a given species were present, a single sequence was printed (the longest). A sequence for each locus was retrieved for the presumed sister taxa *Dascillus cervinus* (Grebennikov and Scholtz, 2004), which was appended to the files.

Loci are separately aligned using the BlastAlign consensus method (Belshaw and Katzourakis, 2005; Hunt and Vogler, 2008). Here, a random subset of 50 sequences are taken from the file, from which the most representative sequence (MRS) is inferred. This is repeated 50 times, then the 50 resulting MRS's are then aligned with ClustalW (under gap opening penalty of 22.5 and extension 0.83) (Thompson et al., 1994). A consensus sequence is made from this alignment by taking the most frequent nucleotide at each position. All sequences for the given locus are then aligned in reference to this consensus sequence, using Blast.

Analysis

We assess the change in phylogenetic signal upon the addition of sequences of a given locus, to existing datasets. The existing dataset (or reference dataset) is generated from a concatenated alignment built from (initially) 16s, 18s and 28s. We wish to know whether adding increasing number of cox1 sequences to the existing dataset leads to change in phylogenetic signal. As response may also be dependent upon both the existing and additional sampling, both existing and the cox1 datasets are resampled prior to combination.

A flowchart giving the study design (appropriate for the analysis of cox1 addition to the existing locus), is shown in Figure 3.3. All steps from resampling to tree inference are looped. The analysis loop begins with the extraction of a subset of the sequences from the reference alignment pool. The proportion retrieved is uniformly distributed (with a

minimum of 100 sequences, with the outgroup always included), and each sequence is given equal sampling probability (random numbers are generated using R: R Development Core Team, 2009). Similarly, I retrieve a subset of the cox1 sequences, which are to be added to the existing (rarefied reference) dataset. The proportion taken from the added locus also uniformly distributed but also independent of the proportion from the reference matrix, such that some resamples will consist of adding a small number of cox1 sequences to a densely sampled existing matrix, some a large number of cox1 to a similarly large existing matrix, and so on. The cox1 sequences when appropriately concatenated to the existing dataset are evaluated against the existing dataset in isolation. I estimate the phylogenetic signal by performing a treesearch on the alignments, and scoring the recovery of expected taxonomic clades. This taxonomic congruence score is calculated by generating an input file with a character matrix giving the taxonomic groups each terminal belongs to, then a parsimony algorithm infers the number of taxonomic state changes on the given tree. The existing alignment is first quality assessed by treesearch and congruence scoring. I then look for change in quality upon the inclusion of sequences from the additional locus. The additional cox1 sequence subset are concatenated to the existing (also subsetted) alignment. The sequences added to the existing matrix may be either matched such that no new taxa are introduced (targeted addition), or random (allowing introduction of new taxa). Both of these options are implemented, with matching (and optional discarding of new taxa) at the concatenation step. Treesearches are run on the matrices using RAxML v7.2.5 (Stamatakis et al., 2005), with a random starting seed, GTRCAT model with 4 rate categories and the –D switch (which stops treesearching when branching distance between successive trees is very low).

Tree quality is assessed both as a whole (ensemble congruence for each rank), and on portions of the tree, by individually scoring each genus, subfamily and family for quality (code in Appendix 1.5). The tree built from the existing dataset is assessed in an unchanged form. Under the random (non-targetted) analysis settings the matrix (when cox1 is combined) contains taxa not present in the existing dataset, so prior to scoring, these additional taxa are pruned from the tree to form a set of terminals equivalent to that

from the tree inferred from the existing alignment. Further details of tree assessment given in chapter two.

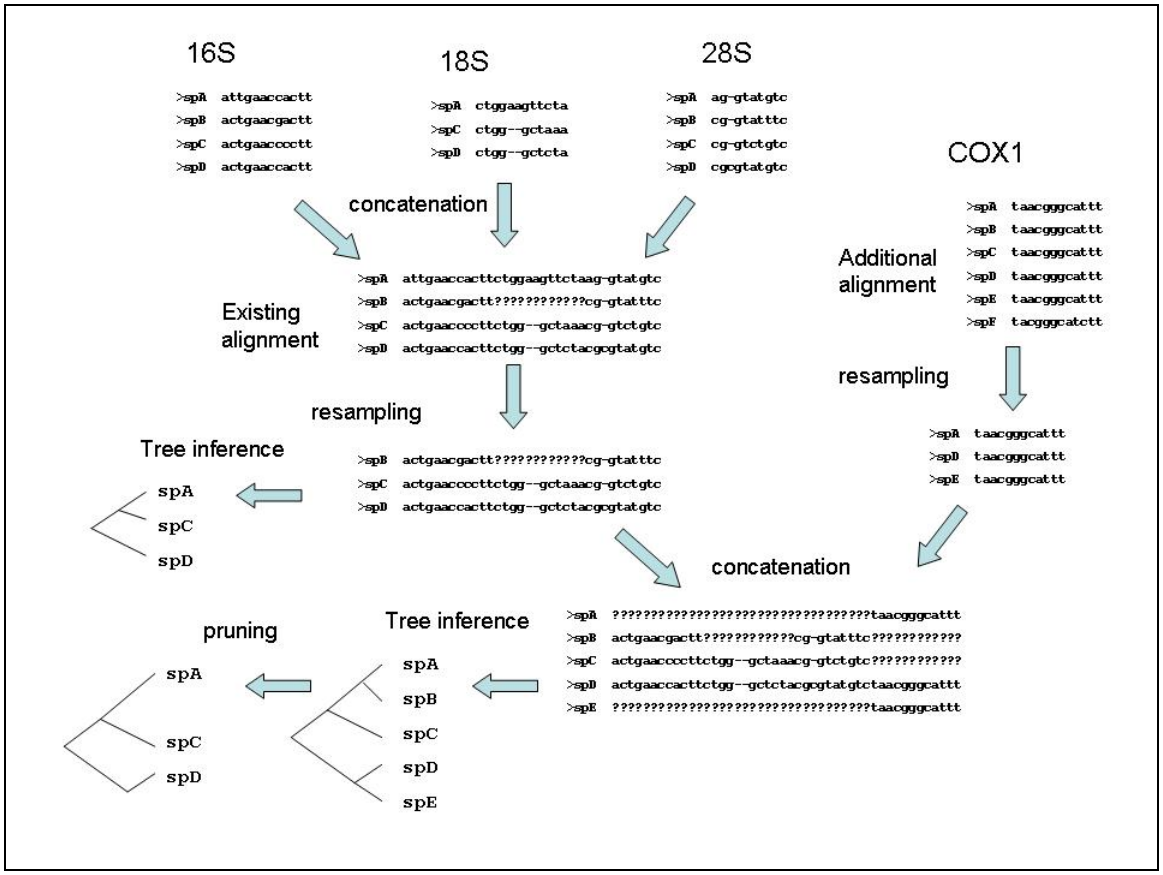


Figure 3.3: Flowchart of the current analysis. An existing supermatrix is formed from concatenated 16s, 18s and 28s datasets. This is resampled and concatenated to cox1 (which is also resampled). Trees with cox1 are pruned to ensure identical taxon sets for score comparison.

3.3: Results

Scarabaeoidea DNA sequences for the four loci were retrieved by homology search from the database and species filtered. After species filtering the cox1 dataset was reduced from 2014 to 788 sequences, the 16s dataset from 840 to 614 sequences, the 18s dataset from 224 to 206 sequences, and the 28s dataset from 842 to 534 sequences. Across the four loci were 1029 named species in total. Of these, 416 were represented by a single locus, 189 by two loci, 348 by three loci, and 76 taxa had all four loci sequenced. Table 3.1 gives the major clades of the Scarab family. Sampling of this locus is currently dominated by the subfamilies Aphodiinae, Melolonthinae, Rutelinae, Scarabaeinae and Sericinae, with cox1 as the most frequent locus. The subfamilies Aesalinae, Syndesinae and Trichiinae are not included in further analyses due to insufficient locus sampling. Reference alignments were built, with the taxa concatenated being those that had a cox1 gene sequence and at least 2 of the three remaining loci. Given previous findings (Wiens 2003), using only relatively complete taxa in this way negates compounding missing data effects. The requirement that a taxon must have the main gene present as in the ideal case each taxon would variously have an additional locus, and not so (and in combination), in order to measure any influence of taxon addition. After such filtering, reference alignments for cox1, 16s, 18s and 28s contained 423, 414, 116 and 401 species, the smaller size of the 18s reference alignment (composed of concatenated cox1, 16s and 28s) being due to reduced representation of this locus.

Subfamily	COX1	18s	16s	28s
Aesalinae	0	3	2	0
Aphodiinae	168	68	42	57
Cetoniinae	16	9	8	9
Dynastinae	9	6	10	8
Hopliinae	5	2	4	4
Lucaninae	5	2	21	0
Melolonthinae	49	32	49	48
Orphninae	1	1	0	1
Rutelidae	8	4	8	7
Rutelinae	44	9	51	45

Scarabaeinae	368	19	293	244
Sericinae	99	26	106	101
Syndesinae	0	0	1	0
Trichiinae	1	0	0	0
Valginae	1	0	1	1

Table 3.1. Current sampling (number of species) of major clades, for four loci.

The reference alignments along with their appropriate added locus files were resampled to assess the impact of adding various amounts of the additional locus to an existing dataset. Table 3.2 gives mean congruence scores at different taxonomic levels for the resampled reference datasets. Score comparison across the reference alignments is not advisable due to variation in taxonomic groups used in scoring, for example fewer 18s genera have multiple sequences (monophyly may be scored when greater than 2 sequences) than cox1. However at the subfamily level the taxon composition is most similar. Here we see tree quality suffers most when omitting 18s from analysis (the lowest tRI is recorded, 0.846), perhaps expected for a slow evolving locus at a relatively deep taxonomic node.

	COX1	16s	18s	28s
Genus	0.780	0.743	0.930	0.756
Subfamily	0.937	0.932	0.846	0.936
Family	0.631	0.611	0.751	0.590

Table 3.2: Mean taxonomic consistency (across resamples) of dataset, where the stated locus (given in row one) has been omitted. For three taxonomic ranks (rows). Across locus comparison is most appropriate at subfamily level (shaded), due to highly similar taxonomic sampling at this level.

The addition of sequences from an extra locus often leads to a change in taxonomic congruence of trees inferred from the matrices, over trees omitting the locus. Figure 3.4 gives the proportion of replicates in which no change in quality is observed upon locus addition. At all ranks we see that when a small existing dataset is used (<100

taxa), the inclusion of an extra locus (of which the number of sequences will range) is less likely to have an impact on the quality of the resulting phylogeny than cases where the existing dataset is more densely sampled.

The cox1 data was added to an existing dataset which ranged in size from ~ 100 to 450. There is some indication that adding cox1 to a dataset at the lower end of this range will give a greater benefit than adding cox1 to an existing dataset which is more complete, as a negative correlation is found between the number of sequences in the resampled existing dataset and the change in tRI upon cox1 addition. However the negative correlation is minor, and significant only at the subfamily level. ($r = -0.049$, p-value = 0.233 for genus, $r = -0.120$, p-value = 0.003 for subfamily and $r = -0.025$, p-value = 0.558 for family level scoring).

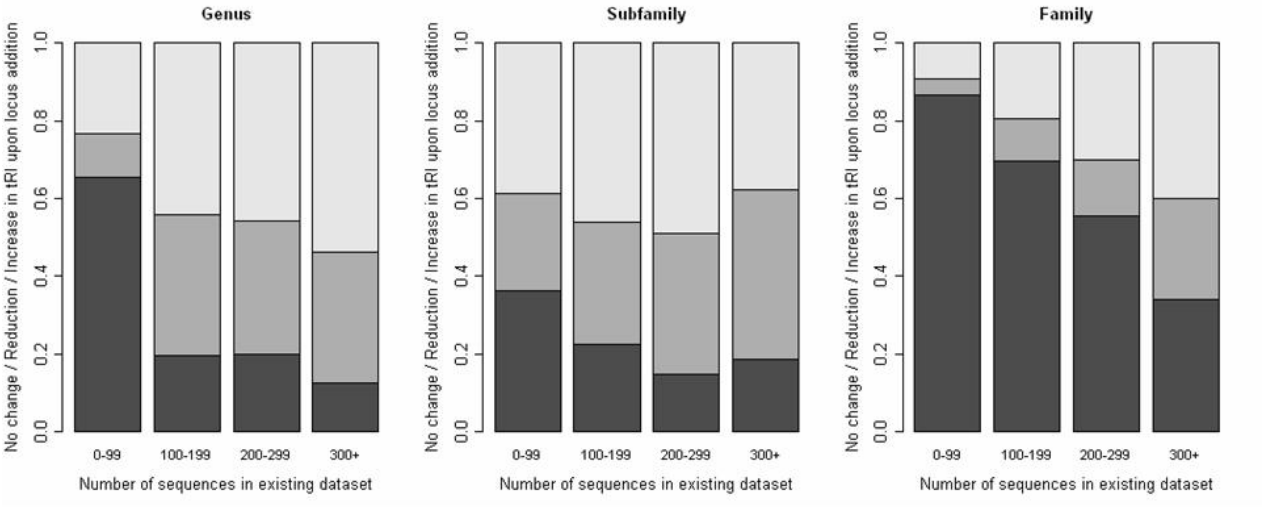


Fig 3.4. Proportion of resamples with no change (black sub-bar), reduction (dark gray sub-bar) and increase (light gray sub-bar) in taxonomic consistency, for different sizes of (existing) dataset. COX1 results are shown.

While the addition of cox1 to a rRNA dataset does not always lead to change in the level of congruence, cases where the matrix with the additional locus do differ, are significantly more likely to be improved (Table 3.3, 'cox1 targeted' rows). Of the 600 additions resulting in a change in generic taxonomic consistency, 0.589 showed an improvement and 0.411 showed a reduction ($p=0.0002$). Similar improvements are seen at deeper levels, with 55% of varying congruence being positive for subfamily, and 63%

at family level. However, the response to locus addition is found to be dependent on the number of sequences added. cox1 results are given in Figure 3.5 show the proportion of taxa in the resulting supermatrix which have a cox1 sequence appended, against the change in congruence. I find a significant correlation between these two variables at the genus (p-value = 0.0002, cor = 0.153, Pearsons product moment correlation), subfamily (p-value = 0.0008, cor=0.137) and family levels (p-value = 0.001, cor=0.138). Where a small number of cox1 sequences are added to the existing dataset (LHS of the graphs) the addition of the extra locus appears to have a negative effect on tree quality, as indicated by the reduction in taxonomic congruence, but as the proportion of taxa with cox1 is raised to 30% or above, all three taxonomic levels tested start to show improvement in tree quality.

	correlation	correlation_p-value	proportion_of_scores_improving	chi_sq_p-value
cox1 targetted genus	0.153	0.0002	0.589	0.0002
cox1 targetted subfamily	0.137	0.0008	0.549	0.0331
cox1 targetted family	0.138	0.0011	0.632	5.1E-05
cox1 random genus	0.09	0.0282	0.539	0.1069
cox1 random subfamily	0.0223	0.5860	0.51	0.6480
cox1 random family	0.017	0.6810	0.542	0.1830
16s genus	0.316	1.55E-15	0.701	<2.2E-16
16s subfamily	0.308	8.22E-15	0.701	<2.2E-16
16s family	0.185	6.75E-06	0.702	4.9E-11
28s genus	0.235	2.47E-09	0.695	<2.2E-16
28s subfamily	0.291	9.15E-14	0.786	<2.2E-16
28s family	0.137	0.0008	0.702	5.7E-09
18s family	0.12	0.0032	0.61	1.4E-06

Table 3.3: Results of statistical tests of the influence of the cox1 sample size on change in quality. Column 1 gives locus, rank and analysis type (targeted and random for cox1, targeted for others). Column two gives the correlation between number of sequences added and the resulting change in tRI, column three gives the significance of this correlation. Out of the resamples in which a change in tRI is observed, column four gives the proportion in which the score from the matrix with the added locus is higher, and column five gives the significance of this (from a chi squared test).

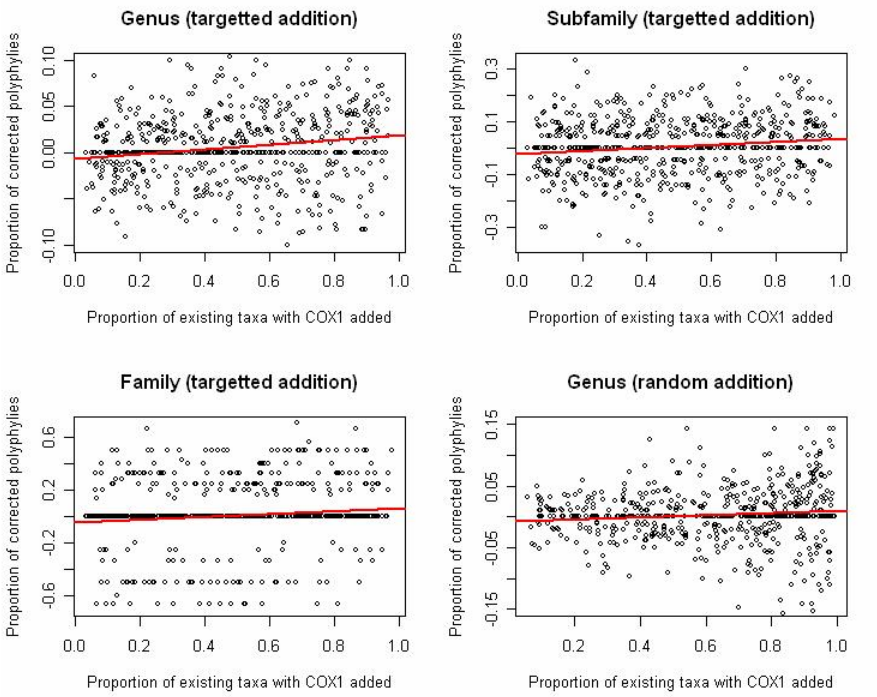


Figure 3.5: Plots showing proportion of taxa with a cox1, versus proportion of corrected polyphylies upon addition of cox1, for tests in which a significant correlation is found

In the above results, taxa from the added loci are targeted to the existing dataset, meaning a given sequence from the additional locus is only included if the taxon is already present in the existing dataset. A second analysis looked at random addition to determine whether the inclusion of sequences from extra loci (where these may introduce new taxa to the matrix) would impact the quality of the phylogenies. Again, I resample the cox1 matrix and add these sequences to the existing dataset. The taxa added are precisely as in the previous analysis, but now allowing the concatenation where a new taxon is introduced to the existing matrix. The trees generated from the matrix with the cox1 added show an increased likelihood of improved congruence than the trees inferred from the existing matrix alone (table 3.3, 'random addition'), but this is generally non-significant. Out of 600 resamples, in cases of variation in congruence, 54%, 51% and 54% at genus, subfamily, and family, respectively, show improvement in taxonomic monophyly upon the addition of cox1, but these are not significant. Although scoring at genus level, the change in quality is modestly correlated (r=0.09, p=0.028, Pearsons

product moment correlation) with proportion of taxa with a cox1 sequence (Figure 3.5, lower right).

While cox1 is of particular interest, it is important to determine whether the results are locus-specific. To this end I perform a further 600 resamples of the 16s, 18s and 28s alignments, where the resamples are concatenated to the existing alignments (of the reaming three loci, including cox1). The addition of random subsets of 16s and 28s loci are found to have a beneficial effect on resulting trees at all taxonomic levels (summary of results given in Table 3.3), with highly significant correlations between sample size and change in monophyly. As discussed earlier, the current 18s results are skewed particularly at genus and family level, but at subfamily level I find significant improvements to the existing dataset upon addition.

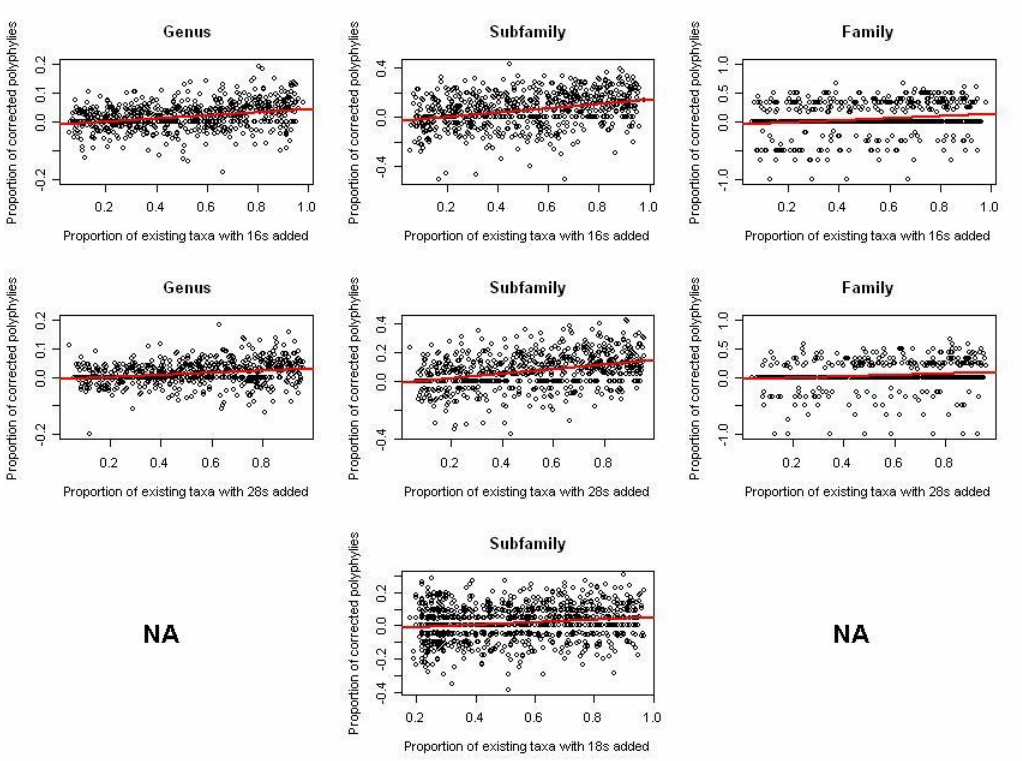


Figure 3.6: As in Figure 3.5, except results for 16s (upper row), 28s (central row) and 18s (lower row) are given here.

3.4: Discussion

There are many facets to the development of the tree of life, but primary amongst these is the data upon which inference occurs. The increase in the amount of data is often discussed, but data structuring is equally relevant, as loci used and taxa sequenced are rarely representative of the genetic and taxonomic diversity. The criteria for locus choice are typically ease and reliability of amplification, primer availability, and often historical reasons such as existing representation. Building the tree of life from such a limited portion of the genome no doubt influences the inference. As the skew to these loci is marked (cox1 in particular), important questions arise: is the signal from the phylogenetic loci accurate and sufficient, which nodes can be resolved through this signal, what sampling structuring is required to pick up the signal, and are these factors consistent for the represented loci. Due to adoption of cox1 in barcoding studies (Hebert et al., 2004), representation of this particular locus will grow like no other. This has led to particular interest in placement of its phylogenetic utility.

The results presented here are of particular relevance in terms of cox1, as they address the impact of this locus as it would likely occur given current trends. The Barcoding life project aims to obtain sequences for all species (Savolainen et al., 2005). Typically these studies are trailed for particular taxa, but assuming no critical problems, ultimately the goal would be to acquire cox1 for the majority of extant species. This steady increase in the representation of cox1 in relation to existing (non-cox1) data means generally limited taxonomic overlap, with perhaps many taxa with existing loci and cox1 but also many taxa represented by only the cox1 locus. Building a matrix structured in this way therefore means a substantial number of taxa with missing data for all but a single locus, but the crucial point being cox1 as the single locus. This corresponds to the random sample regime here, and we see this random inclusion of this locus leads to little improvement in phylogenies. However, the correlation of number of sequences with improvement at genus level is particularly relevant. As sampling tends towards high density, even random addition will lead to improved phylogenies at the tip level. Whilst the tree is inferred of a matrix with additional taxa, as these taxa are pruned for scoring (see methods) they do not contribute to congruence scores, meaning it is the overlapping

nodes that are benefiting from the increased data. Under the sampling regime implemented, resamples with a large number of new cox1 taxa will tend to be those also with a generally large proportion of the existing taxa represented. This of course representing a real-world scenario. However it does mean that it cannot be inferred that the a large number of new taxa are more substantially benefiting the resolution of the existing nodes, as it is more likely that the cox1's concatenated to existing taxa are doing this. But that the presence of new taxa does not prevent or outweigh [maybe by being deadweight] any benefits to the existing taxa. I have not determined whether the new nodes added on inclusion of cox1 are informative in their own right, but it is likely that the high degree of missing data at these taxa (i.e. all loci missing except cox1) would be a source of inaccuracies (Wiens, 2003).

Perhaps surprising were the findings that deeper taxonomic nodes showed significant improvements from cox1 inclusion when addition is targeted (Table 3.3, rows cox1 targeted subfamily, family). The rapid evolutionary rate in cox1 results in homoplasy, particularly at higher ranks. However the lower rate at the 1st and 2nd positions can only make homoplasy less prevalent in these partitions. Still, the homoplasy and saturation has not been found to swamp the phylogenetic signal, as deep family nodes are still improved upon inclusion of cox1. The results are encouraging when considering the role cox1 will play in the tree of life, the adoption of this locus will most likely give improvements to existing nodes, an effect that is significant at a substantial evolutionary range. Although the trends of increased phylogenetic accuracy (as determined here) appeared to differ across the loci tested, with number of rRNA's added more strongly correlated with improvement in quality. Parameters such as substitution rate, alpha, proportion of invariant sites, and the number of characters, vary for cox1 and the ribosomal genes. Using the current accuracy measure, these variables must influence the change in quality upon locus addition, most likely the increased homoplasy in mt loci (Danforth et al., 2005) is responsible for the decrease in phylogenetic performance at the deeper levels.

The required degree of taxon sampling has received much coverage in the literature, but robust analyses of empirical data are in short supply. The approach here overcomes previous limitations imposed due to the use of single loci. Typically varying

numbers of taxa are used to infer trees, which are then assessed for quality. In the case in simulation this would be some measure of topological distance to the true tree (the topology upon which the sequence evolution was simulated), or alternatively distance to some independently derived answer (such as taxonomic / morphologic / molecular congruence). Often as the number of taxa used is increased, the similarity between the resampled tree and the independent tree increases. While this information is useful in terms of advising phylogenetic methodology, it is limited in that the most that can be concluded is the benefit of relatively increased sampling, the results lack a benchmark. Here a benchmark is included by assessing the performance relative to the existing dataset. The existing 16s/18s/28s supermatrix attains a specific degree of similarity to the ideal (a specific similarity to the independently derived ideal), this attainment may be aided or worsened by specific numbers of cox1. In the absence of this benchmark, typically one would compare subsamples of the gene to the complete sample, testing whether progressively fewer numbers has any impact on the score. The limitation being it fails to inform us whether the maximal score (usually from the complete sample) is good in itself.

Inspection of the plots appears to suggest that sampling a small proportion of cox1 may lead to a worsening of the trees. In the case of cox1 it will get increasingly unlikely that this lack of taxonomic overlap would occur, for most taxonomic groups. But of course this would be a factor for many other loci. Why the dataset suffer in quality, is uncertain. Previous authors have discussed the tolerance to missing data (Wiens, 2003), and found inaccuracy prevalent where missing data occurs below a threshold. In the current case, it is not simply that the addition of the extra data, is providing no further beneficial information or phylogenetic signal, but appears to be actively misleading the search. We can only hypotheses about the mechanism, but homoplasy, long branch attraction, and perhaps the computational burden of additional sequences (with little phylogenetic signal) are candidate causes. Still, in light of the discussion above, whilst 80% cox1 missing data may give a higher score than 90%, both would be counterproductive in an attempt to improve a data matrix. It is reiterated that there would be few cases in which such a small number of cox1 sequences would be available relative to existing data. But for general locus addition this needs taking into account. Of course

the situation is more complex when there is number of additional loci which may be used. I have shown here problems when taxonomic overlap of new to existing data drops below 20%. But were a new dataset published, high taxonomic overlap may be present in the newly available sequence, whether or not these also overlap with the existing data. This question would perhaps require multi-dimensional resampling (i.e. loci plus sequence numbers), and is beyond the scope of the current paper. Here I focus on improving an existing dataset with barcode data.

Chapter Four: Generation of Supermatrices From Sequence-Based Species Delineated Loci.

4.1: Introduction

Generating a supermatrix from mined data.

The growth of available sequence data mean the use of multiple loci in phylogenetic analysis is now ubiquitous, with two general approaches to incorporating the additional data into subsequent analysis. These are, the combining of trees built from individual loci (source trees, or gene trees) into a single species tree, or the concatenation of data from individual loci into a single matrix (supermatrix), on which the tree search is performed. In an ideal case, each concatenate in a species level supermatrix would represent a single sequence for each gene, for each species. Limitations imposed when using mined data present difficulties in forming species level concatenates. Mined data often has a substantial proportion of sequences without species level identification (Nilsson et al., 2006). Sequences from taxa lacking easily diagnosable traits such as bacteria and fungi are commonly submitted with incomplete taxonomic information, typically with a specimen tag assigned to the species field. Building a species level matrix of unidentified data in these cases requires prior species assignment, typically achieved by comparison of the sequence to a library of identified sequences. A species name may be assigned if the unidentified sequences shares sequence similarity to, or monophyly with, a named sequence (Drancourt et al., 2000; Hebert et al., 2003; Ryberg et al., 2008; Brock et al., 2009).

Creating species concatenates of mined data is further complicated as multiple alleles are typically available for many species. Where species information is available alongside sequence entries, filtration and concatenation is a matter of discarding all but a single sequence entry for each species tag (species name where the specimen has been identified, or specimen label where unidentified), then taxonomically filtered species data can be combined by matching identifiers between loci followed by appending sequence, for each successive locus. The problem of missing data can be dealt with at this step, missing data characters are inserted into the growing concatenate (‘?’ or ‘N’ are

commonly used) where data is not available for a given species. The compilation of the matrix then becomes a trade off between maximizing the data completion per species, and the absolute number of species (Sanderson et al., 2003).

Molecular species delineation

While the species labels in databases are morphologically derived, the assignment of individuals into discrete groups need not necessary use this information. There is ongoing work on the delineation of species boundaries based on analysis of the sequence information alone. A sub-group of individuals may be considered an independently evolving lineage if they share fixed nucleotide differences (population aggregation analysis and cladistic haplotype analysis, Davis and Nixon, 1992; Brower, 1999), if the clade is restricted to a population (Wiens and Penkrot, 2002), if the molecular variation is below a defined threshold (the barcode approach: Hebert et al., 2004), or the branch pattern more closely equates to a coalescent process (General Mixed Yule Coalescent Model: Pons et al., 2006). Evolutionary theory gives us a number of reasons that species boundaries based on morphological traits may not be congruent with those based on sequence variation (Ballard and Whitlock, 2004). Cryptic species lack morphological variation from their sister species, but are revealed by the increase in sequence variation between them. Conversely, hybridising sister species results in homogenous sequence, particularly at mitochondrial loci, but morphological differences remain. In both these cases, assigned species labels would not be expected to match sequence-delineated species labels. Similarly, the inferred evolutionary history of a given locus may differ to that at other loci. The fixation of alleles at different loci will not precisely match the species lineage, leading to variation in branching patterns in inferred relationships (incomplete lineage sorting). These causes of incongruence are compounded by a number of evolutionary patterns which lead to incorrectly inferred phylogenies. Heterogeneous evolutionary patterns across loci (e.g. AT bias and substitution rate variation) lead to characters potentially misleading to treesearch algorithms, and incorrectly inferred histories (Tateno et al., 1994; Yang, 1996a). The issue of incongruence in molecular species delineation has yet to be addressed.

Concatenation of sequence filtered loci.

Concatenation of taxonomic species filtered loci is a simple case of matching names from the different loci, then tagging one string to the end of the existing sequence. Similarly, where specimens are used, the same method would apply. However, loci filtered by sequence-based approaches present a challenge. If it is assumed that each sequence within a sequence boundary (along with its taxonomic name) belongs to the same sequence-delineated 'species', then the entity may encompass multiple taxonomic names, or a given taxonomic name may be split across multiple sequence-based species. Due to incongruence, there may no longer be a one-to-one correspondence across the loci, as it is unlikely that the resulting clusters precisely match the established taxonomy. Further, heterogeneous patterns of sequence evolution across molecular datasets mean that delineated species may not match between loci.

While only a single sequence in the cluster is used in concatenation, this choice is little more than arbitrary. The taxonomic label attached to the single sequence must somehow take into account the taxonomy of all (including the filtered) members of the cluster. The attachment of multiple species names to a single sequence results in concatenation choices; these are, situations where a growing concatenate may match multiple clusters in additional loci. This situation also results from cases when a taxonomic species name is spread over multiple clusters. For large scale datasets, isolated cases could perhaps be acceptably resolved by random concatenation. However, additional incongruence soon results in compounded complexity, where random concatenation would likely lead to suboptimal situations as initial loci concatenations impound the taxa involved, this leads to the necessitation of more missing data. Here I address the issue of concatenating sequence filtered loci, in a large multi-locus order-scale dataset. Using a purposefully rough-and-ready molecular-species clustering approach, I generate mOTU's showing a substantial degree of incongruence both with the taxonomy, and across the loci. I use this dataset to demonstrate a search method for a combination of concatenates that maximally match the established taxonomy.

4.2: Methods

Database and taxname generation (code in Appendix 1.1)

The Genbank taxonomy database was downloaded from ftp://ftp.ncbi.nih.gov/pub/taxonomy/ . NCBI give a unique code for each taxa. This is contained in the names.dmp file, along with the corresponding nomenclature, synonyms, common names, blast names, Genbank common names, common misspellings, equivalent names and part names etc. The nodes.dmp file contains hierarchical information giving taxonomic relationships. The nodes.dmp file was first parsed. Each line in this file represents a single node in the taxonomy tree, containing the code representing the node itself, along with its parent node (also has rank). Once the node structure and scientific names of each node are stored, the tree is traversed in a recursive manner to create unique species codes (Hunt and Vogler, 2008), and generate a keyfile giving taxonomic information for each species. Table 4.1 gives an example from the Scarabaeoidea starting node (Genbank id = 75546). This node contains thirteen child nodes (families), the first of which is Scarabaeidae. The existing taxcode (for the parent node) along with this child node name is passed into a subroutine which checks the rank of the node. For most internal ranks, a single name is given for the node, so this string is walked in order to create a code for that node. However at species level the name string can be composed of multiple individual words separated by whitespace. In the ideal case of two words (most likely a binomial), the second word is extracted. In some cases three or more words are present so taxonomy terms such as sp aff nr cf spp BMNH are ignored here. This word is passed into a subroutine which takes progressively larger substrings of the name until a unique name is generated (names are check via hash tabling). Once a unique code is found for the given node, this is returned and it progresses to any child nodes (grandchild nodes of Scarabaeoidea), if any. As the traversion command is called from within the subroutine itself, it proceeds through all child nodes (a common method of dealing with tree-like structures).

The Genbank invertebrate release (DNA) flatfiles were downloaded from NCBI and each parsed for the DNA sequence, accession and NCBI tax id. Sequences containing Coleoptera NCBI taxid's were filtered from the files and assigned the matching taxstring,

making the final Coleoptera database. The 'formatdb' NCBI program (ftp://ftp.ncbi.nih.gov/toolbox/FAQ.html) was used to generate a blastable database from this fasta file. The Coleoptera database was searched in the same way as the Scarabaeoidea database in chapter three (section 3.2), but without name based species filtering.

taxstring	name	rank	family	subfamily	genus	species
SS4	Scarabaeidae	family	Scarabaeidae			
SS4C5	Cetoniinae	subfamily	Scarabaeidae	Cetoniinae		
SS4C5Pa7	Pachnoda	genus	Scarabaeidae	Cetoniinae	Pachnoda	
SS4C5Pa7mar	marginata	species	Scarabaeidae	Cetoniinae	Pachnoda	marginata
SS4C5Os7	Osmoderma	genus	Scarabaeidae	Cetoniinae	Osmoderma	

Table 4.1: Taxcode generation example. A taxstring is created for each taxon found in the Genbank database.

Species filtering

Building a species level phylogeny via the supermatrix approach requires the concatenation of aligned loci for multiple individual species. Prior to alignment and concatenation, the species must be defined separately on a locus by locus basis. Many species will have multiple sequences present on the database, so it is necessary to filter these, leaving a single sequence for each species. The computational demand of aligning order-scale datasets is substantially lowered when unnecessary intraspecific data is filtered, so it is desirable to perform species-filtering *prior* to alignment. The species filtered dataset may then by aligned and combined with the corresponding species for the other genes. Species filtering is typically performed using assigned names (following morphologically-based taxonomy), as each sequence has a species id attached. There have been recent attempts to define species limits based on DNA sequence variation, assuming that variation between species is much greater than that within a species. These sequence-based species definition methods are in their infancy, and their development typically on small scale, aligned matrices. Their application to order-scale datasets is difficult; therefore in this instance I opt for a basic Blast-based clustering method to define species limits (Brock et al., 2009). Each similarity-based cluster is assigned a

name, based on the taxonomically-assigned species names existing within the cluster. Where a cluster contains multiple species names, the most commonly-occurring is used, with a number appended to give the number of additional species names. This dataset is filtered to leave a single sequence per cluster, then the sequences aligned and concatenated.

For each locus, BlastClust was invoked under a number of different similarity cutoffs (at a given similarity cutoff, sequences are clustered when the percentage of identical sites is greater than this value). The clusters contained in the BlastClust output were then analyzed in terms of the taxonomically assigned species names they contained. I count the number of correctly defined species (the number of cases where a taxonomic species with more than one sequence has all its sequences in a single cluster, with the exclusion of sequences from other taxonomic species) for each BlastClust cutoff. Unidentified species (those without the typical binomial format id) were necessarily ignored.

Concatenation

Clusters whose representative sequences could be easily concatenated were first identified and removed from the files. These are cases where a species id is only present in isolation at individual loci (not clustered with a different species) and where species are present only at individual loci. The remaining species are those duplicated or combined, in at least one of the loci. For these, there exists a number of concatenation options, which can be visualized as paths through the sequence x loci matrix (Figure 4.1).

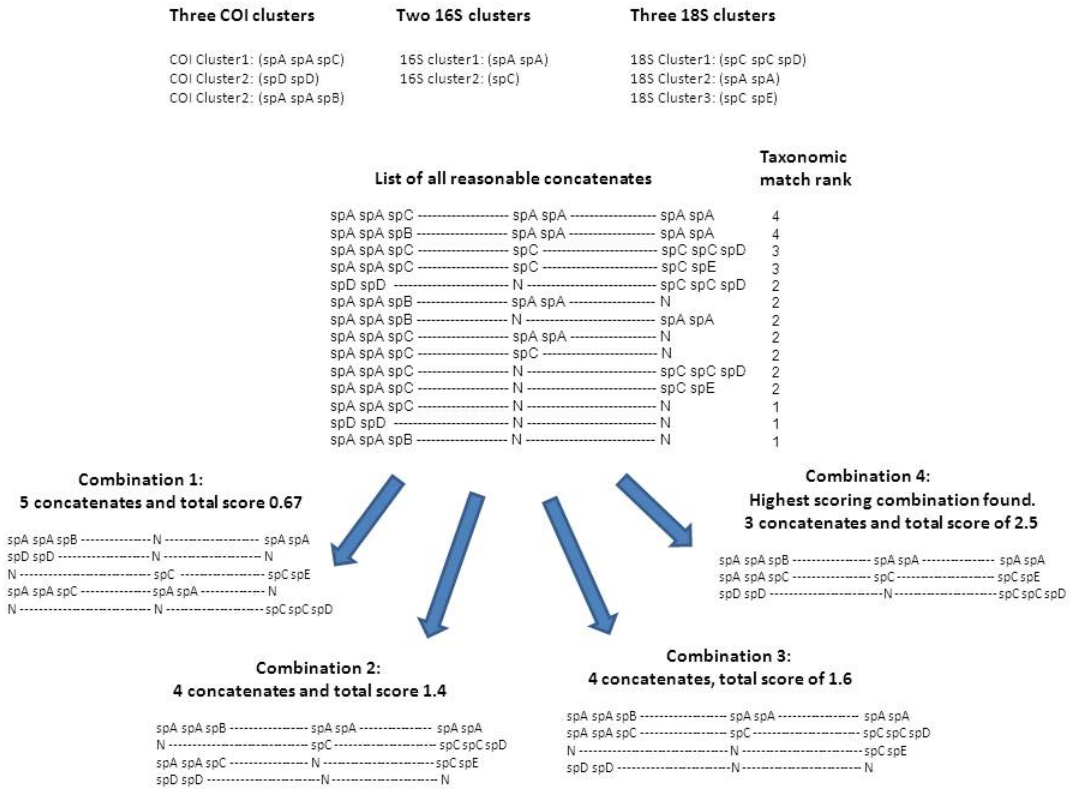


Figure 4.1: Illustrating alternative ways of concatenating incongruent clustered loci, necessitating a search for an optimal combination. Three loci (COI, 16S, 18S) have been clustered, for example the eight COI sequences have formed three sequence based clusters (cluster 1 is composed of two sequences from species A and one sequence from species C, cluster two is composed of two sequence from species D and so on). A single sequence is taken from each cluster to be concatenated to the other loci, or to missing data (N). Concatenation is represented by hyphenated lines. All reasonable concatenation options are listed (central), then this list is sampled a limited number of times, finding (in this case) four differently scoring combinations.

The vast majority of concatenates may be ignored, as most will produce complete taxonomic mismatches. To traverse reasonable concatenates, I start at a given cluster in loci 1, and then obtain lists of all taxonomic matches at other clusters in the remaining loci. Given these lists, I find all possible ways in which the cluster in locus 1 may be concatenated to remaining loci (via recursive traversal through the remaining loci, from the starting cluster in the initial locus). These possibilities are termed potential concatenates. This is then repeated for each start position (each cluster in the first locus). Each reasonable concatenate is scored for taxonomic matches, with the score derived from a pairwise comparison of all clusters in the concatenate. For default scoring I increment the score by two if all members of one cluster are the same taxonomic species

as all members of the other, and by one if a subset of the sequences in one cluster have a matching species name to species in the second cluster. Where there is a specimen match an additional incrementation of one is made. For each locus there is a single additional path option of missing data, in which case the score is not incremented. As completely mismatching paths are not considered, mismatch scoring is not necessary. With this scoring scheme, a concatenate through a 4 gene matrix giving complete species matches would score 12.

The paths and scores are written to file for analysis in R (R Development Core Team. 2007). Given all the reasonable potential concatenates of an order-scale dataset, the number of ways these may be combined is astronomical. As it would not be possible to assess all possible concatenate combinations, I randomly sample the space, attempting to maximise the total score and minimise the total number of concatenates. Two methods for searching the cluster combination space are used. The first method is a random permutation of the concatenate list; a permuted list of all potential concatenation paths is scanned sequentially, then each successive concatenate is stored if all individual clusters in the path have not been stored previously. When the path list has been scanned, the total score (sum of score for each cluster combination / total number of concatenates) for that concatenation combination is calculated. This procedure is repeated many times, and the highest cluster combination used. A second approach searches a limited but high scoring region of the concatenate combination space. This is achieved by ranking the individual concatenation paths by score. A permutation then uses this list of ranks as probabilities, such that higher scoring paths will have a tendency to be retained at the top of the list, and lower scoring concatenation paths will be lower in the list. The permuted list is then scanned as before, to calculate the cluster combination score. In this way, high scoring (those with many taxonomic matches across loci) cluster concatenates are favoured. The highest scoring path is concatenated, using a single sequence from each cluster (the most complete).

Treebuilding

A phylogeny is inferred from the cluster-filtered supermatrix. I first filter the matrix to remove taxa with missing data. Estimates of the allowable level of missing data vary (e.g. Qiu et al., 1999; Murphy et al., 2001; Bapteste et al., 2002; Wiens, 2003), however a combination of mitochondrial and nuclear rRNA is thought to give good resolving power for nodes at many levels in insect phylogentics (Danforth et al., 2005), and as two of the four loci (cox1 and 18s) would fill this requirement (with the advantage that both are well sampled in the Coleoptera), I use only concatenates that contain both these genes. The remaining concatenates are removed. I then perform ML treesearches using RAxML v7.2.5.

For comparative purposes I build an equivalent supermatrix from taxonomic filtered sequences. In this instance, the original retrieved blast hits are filtered to leave a single sequence per species id (species id's include named and unidentified tags). The filtered sequences are aligned in the same way as previous, then concatenated in the standard manner. A second ML phylogeny is built for the taxonomic-filtered supermatrix.

4.3: Results

The Genbank taxonomic database was traversed from the Coleoptera node. Unique taxa were recorded at all ranks from this node, containing 10503 species id's, 7712 of which are binomial and 2791 unidentified. These 2791 unidentified labels contained about 180 partial identifications (cf., nr. or aff.). It is not known how many species these unidentified id's equate to, but it is possible to make an estimate on a locus by locus basis.

BlastClust performs a pairwise blast search of all sequences in the dataset. Sequences are added to an existing cluster if the blast similarity is above a specified threshold. The similarity threshold is set to a given sequence identity percentage by the user, then this value used for all pairwise comparisons in the dataset. Evolutionary characteristics (particularly, rate) are of course highly locus specific (Danforth et al., 2005; Rokas et al., 2002), which means the degree of sequence variation within a species is likely to differ across the loci. I attempt to find a sequence similarity cutoff that results in clusters with minimal deviance from the establish taxonomy. A range of clustering cutoffs are assessed for their congruence with taxonomically defined species. For each cutoff I count the number of identified taxonomic species (represented by >= two sequences) which are clustered to the exclusion of sequences from other species. Blast similarity cutoffs are varied from 90 to 100 in steps of 0.25. Figure 4.2 plots the number of correctly defined species against similarity cutoff, while table two gives details on the optimal cutoff values for each locus.

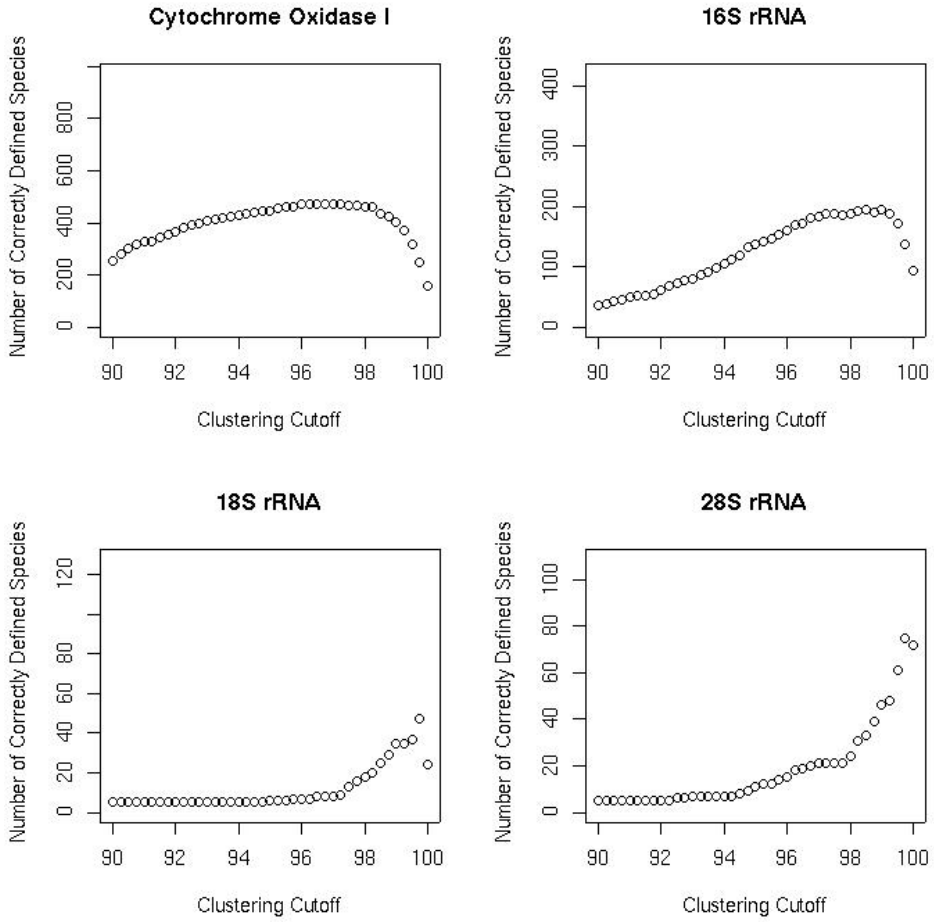


Figure 4.2: Scatter plots giving BlastClust cutoffs versus the number of correctly defined species, for all Coleoptera sequences. Eight loci are shown. Graph y-axes are scaled to the number of species included in each test. For example, of the ~120 testable 18S species, ~60 were recovered at the optimal cutoff..

I find the slowly evolving loci (18s and 28s) return a peak number of correctly defined clusters at almost identical sequence similarity, while the faster evolving mt genes require around 96%. The degree of incongruence of the BlastClust approach with the established taxonomy is quite apparent, only ¼ - ½ of taxonomic species are being recovered, even at these optimal cutoff values.

The clusters generated from the optimal similarity cutoff values were analysed for the assignment of unidentified sequences to clusters containing named species (Table 4.2). Firstly, the number of clusters containing unidentified sequences gives an estimate of the species diversity represented by this class of sequences. In cox1, the 1282 unidentified sequences are clustered into 1023 mOTU's, with similar proportions for the

other loci. When clustered with named species, the unidentified sequences are effectively assigned a species name. However this on occurs in the minority of cases. 147 of the unidentified cox1 sequences are contained in a cluster that also contains an identified species. There are 88 of these mixed clusters in cox1, so the 147 sequences are being assigned to a number of different species. Again, the proportion is similar across the loci, with roughly 10% of unidentified sequences assigned to a named species. This species assignment is improved by comparing species id's across loci. When multiple loci are sequenced for a given specimen, the unidentified species id is shared at multiple loci. This means that if an unidentified sequence is clustered to a named sequence at any locus, other loci also containing a sequence from the same specimen may also be assigned to that same species. The final row in table two gives the number of additional unidentified sequences that are assigned a species label by this method of across locus assignment. While only 147 of the 1282 unidentified COX1 sequences are assigned a species at this locus, a further 90 can be assigned by reference to the clusters at the other loci.

Locus	cox1	18s	16s	28s
# clusters	4760	2092	3260	1129
# clusters all members identified	3737	1366	2517	702
# clusters all members UNidentified	935	674	688	399
# clusters mixed	88	52	55	28
Total binomials	3577	1482	2695	824
Total unidentified sequences	1282	800	1023	553
# of assigned unidentifed sequences	147	75	93	77
# of NONassigned unidentifed sequences	1188	727	987	499
Cross locus assignment	90	18	119	53

Table 4.2: Species assignment to unidentified sequences via clustering

When assessing the reliability of a given result, it is common to test for the existence of a similar signal from different sources. The pairing of two sequences into a cluster in one locus can be considered more reliable if these same two sequences are clustered at other loci. With the current dataset, this is possible in two of the three classes of sequence. The pairing of an unidentified sequence (which has a species id) to another sequences (whether identified or not) can be matched at other loci, using the id as a label. As can the

unambiguous pairing (or otherwise) of two different species. But where multiple identified sequences are available for the same species, lack of specimen labelling prevents matching to equivalent sequences (if any) at other loci. I test for congruence in such clustered sequence pairs. For each locus I retrieve all pairs which are both present in a cluster, and not found to be also separated in clusters elsewhere (which may be termed, uniquely clustered). Then for each of these pairs I count the number of cases where the pair is also uniquely clustered at another locus (agreement, or congruence), and the number of cases where the pair is uniquely separated at other loci (disagreement, incongruence). Table 4.3 gives the congruence counts. I find substantial congruence for unidentified sequences, the clustering of two unidentified sequences at a given locus very likely to be similarly clustered at other loci. Although it is noted that these datapoints are dominated by a small number of large scale studies, in particular Cicindela and Copelatus. For identified sequences, the pattern is less clear, with modest congruence at the fast evolving loci, but general incongruence for the slow evolving genes. As such, the assignment of sequences from different taxonomic species into a mOTU's using BlastClust has poor support. BlastClust is likely too crude for general adoption as species filtering, where tax info is available, but as discussed elsewhere, the accuracy of the method is irrelevant for the current purpose.

	Identified		Unidentified	
	agreements	disagreements	agreements	disagreements
cox1	96	5	5162	38
16s	100	21	5158	29
18s	19	23	31	2
28s	24	49	65	38

Table 4.3: congruence in sequence clustering between loci. Details are given in the text. Note the unidentifieds column is actually including unidentifieds, i.e. these will definitely have an unidentified. But might also have some identifieds.

Clusters that could be easily concatenated were filtered. The clusters retained were i) those which contained a species that was dispersed amongst several clusters and the same species was present at another locus, ii) those which contained multiple species which where the species were dispersed across clusters at least one other locus. The

number of retained clusters were 670 (cox1), 221 (18s), 477 (16s), and 150 (28s). The filtered clusters were concatenated using basic species name string matching (cox1:4090, 18s:1871, 16s:2783, 28s:979).

I mapped all reasonable concatenates via matching or partial matching of cluster members through the [loci X cluster] matrix. 8355 paths were found with at least a partial match (this number additionally includes clusters without concatenation, that is, the path via missing data option). The histogram in Figure 4.3 shows score frequencies for the mapped concatenates. I note that potential concatenates of four loci include a greater number of partial matches, whereas concatenates of two or three are predominantly complete matches (e.g. compare bars 1 (mismatch) and 2 (match)).

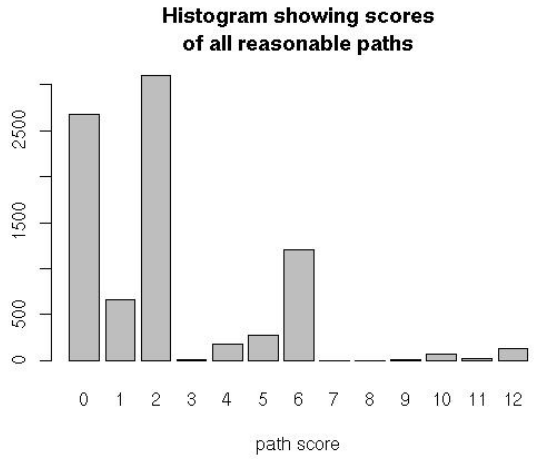


Figure 4.3: Frequency histogram of taxonomic match scores of the 8355 reasonable concatenation paths.

The list of potential concatenation paths was read into R, then a search performed for a high scoring concatenate combination. The space of cluster combinations was searched by 10,000 permutations of the potential concatenate list, using two methods (Figure 4.4). Method 1 which limits permutations within a set of similarly scoring concatenates attained a score of 1.966 producing 1451 terminals (concatenates) after 1160 permutations, whereas the second permutation method produced much lower scores. As the combination space searched by method 2 is much larger, it is assumed it would eventually find a cluster combination scoring higher than method 1, but the graph show this would be unlikely in any reasonable timeframe, using the current implementation.

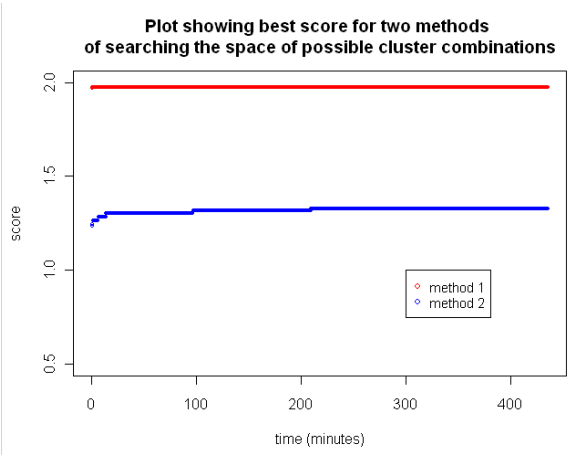


Figure 4.4 : A scatter plot of total concatenate combination score vs time, during a cluster combination search of a large (method 2) or reduced (method 1) combination space.

A supermatrix was generated from the highest scoring path combination. For each path, I take a single sequence for each locus, in cases where a cluster contains more than one sequence, the most complete sequence was retrieved. The sequences were concatenated and a new id created for each sequence. The id used was the majority species name, with '+[number]' appended, indicating the number of additional species names encompassed. The relationship of these new sequence id's to regular taxonomy is illustrated in Figure 4.5. The optimal combination is given in the left column, and a lower scoring combination (from the second search method) is given on the right. It is immediately apparent from the decrease in height of the optimal combination on the left (both are drawn to scale), that the improvement in score of a combination accompanies a reduction in the total number of concatenates. This is entirely expected, as the observation of test datasets (see methods for example) showed unnecessarily high use of missing data. There are many cases in which a sequence based cluster contains multiple taxonomic species. But as can be seen from the graphic, the additional species are close relatives. There are rare exceptions to this rule. For example the E3Eu4Me7bup cluster concatenate contains an expected E3 species, but also from a separate superfamily, Cu3. These most likely represent labelling errors. For the clusters in the concatenation analysis, clusters crossing species, generic, subfamily and family boundaries number 244, 55, 13 and 4, respectively.

For comparative purposes I create a concatenated supermatrix from loci that have been species filtered according to species labels (including species names and unidentified specimen labels). The inclusion of unidentified sequences make for a larger supermatrix when name-filtered, with 7170 Coleoptera taxa, as opposed to 7009 when sequence-based filtering is applied. The resulting supermatrix size is not significantly altered by changing score parameters. When down-weighting or up-weighting partial taxonomic matches, I only find very minor increase in the number of mOTU's (7011 when partial match score 1, and 7012 when partial match is scored as complete match). The amount of missing data in all Coleoptera supermatrices is high. When reduced to only taxa with cox1 and 18s loci present, 856 remain (767 for the supermatrix based on sequence filtered loci). The impact of sequence based filtering in taxonomic representation in terms of species diversity is indicated in Figure 4.6. The frequency histogram shows a reduction in extreme values when sequence filtered. Both supermatrices are dominated by genera with a small number of species, But the hyperdiverse genera (those with >= 200 sequences) are reduced when sequence filtering is applied. A treesearch performed on the sequence-delineated supermatrix resulted in the phylogeny shown in Figure 4.7 (upper). In spite of the reduced size of the sequence-delineated matrix, it is found to be quite comparable in terms of quality to the taxonomic filtered tree (Figure 4.7, lower), giving an identical taxonomic consistency score (tRI = 0.872 whether sequence filtered or taxonomic filtered). This suggests sequence filtering may be more efficient at removing non-informative sequences.

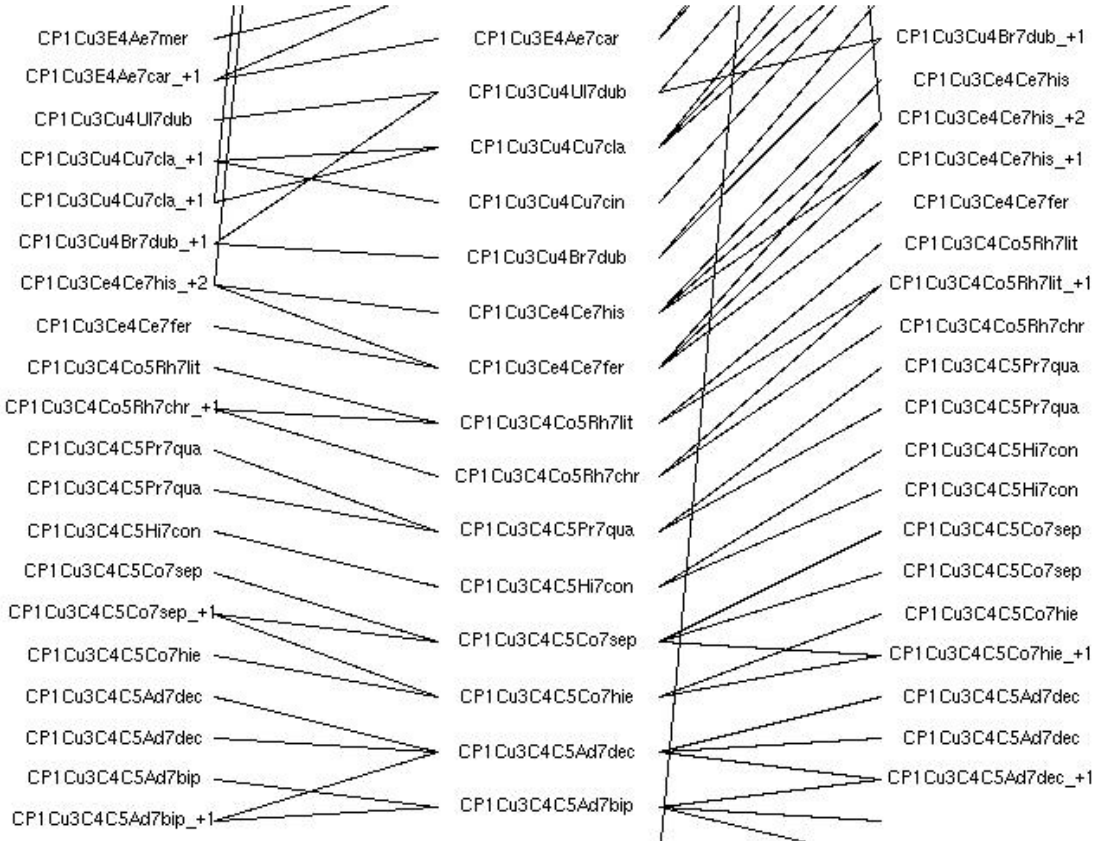


Figure 4.5: Illustrating the limited taxonomic range in which additional morphospecies are incorporated into given sequence based clusters. The central column gives a list of Linnaean taxonomic names for X genera in the Cucujoidea family. The left (scoring regime 2, X) and right (scoring regime 2, -2) columns list the labels for the *concatenated* clusters, for this family. The concatenate labels represent the majority species name followed by +[N], where N refers to the number of additional species encompassed in that concatenate. The figure illustrates that the majority of cases in which multiple taxonomic species (indicated by sp_name+N) are incorporated into a given concatenate were members of the same genus, indicated by lines that deviate little among the columns. However, there are exceptions, as for example the E3Eu4Me7bup (*Melasis buprestoides*; Elateriformia Eucnemidae) concatenate that contains an expected Elateriformia species, but also a sequence assigned to a distantly related superfamily, Cucujoidea. Such extreme cases likely represent labelling errors in the database entries.

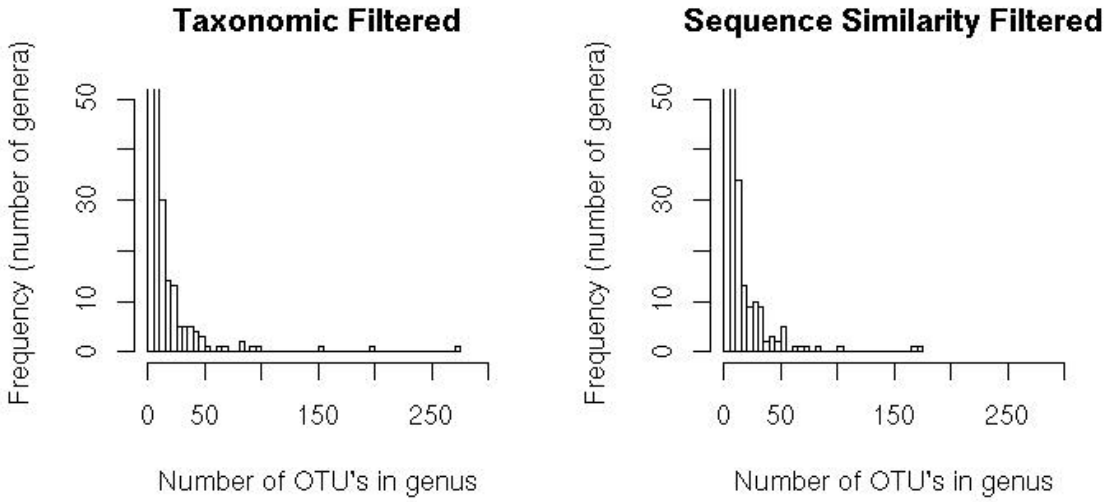


Figure 4.6: Species diversity in the name-based filtered and sequence-based filtered supermatrices. Graphs are frequency histograms for the number of species in a genus.

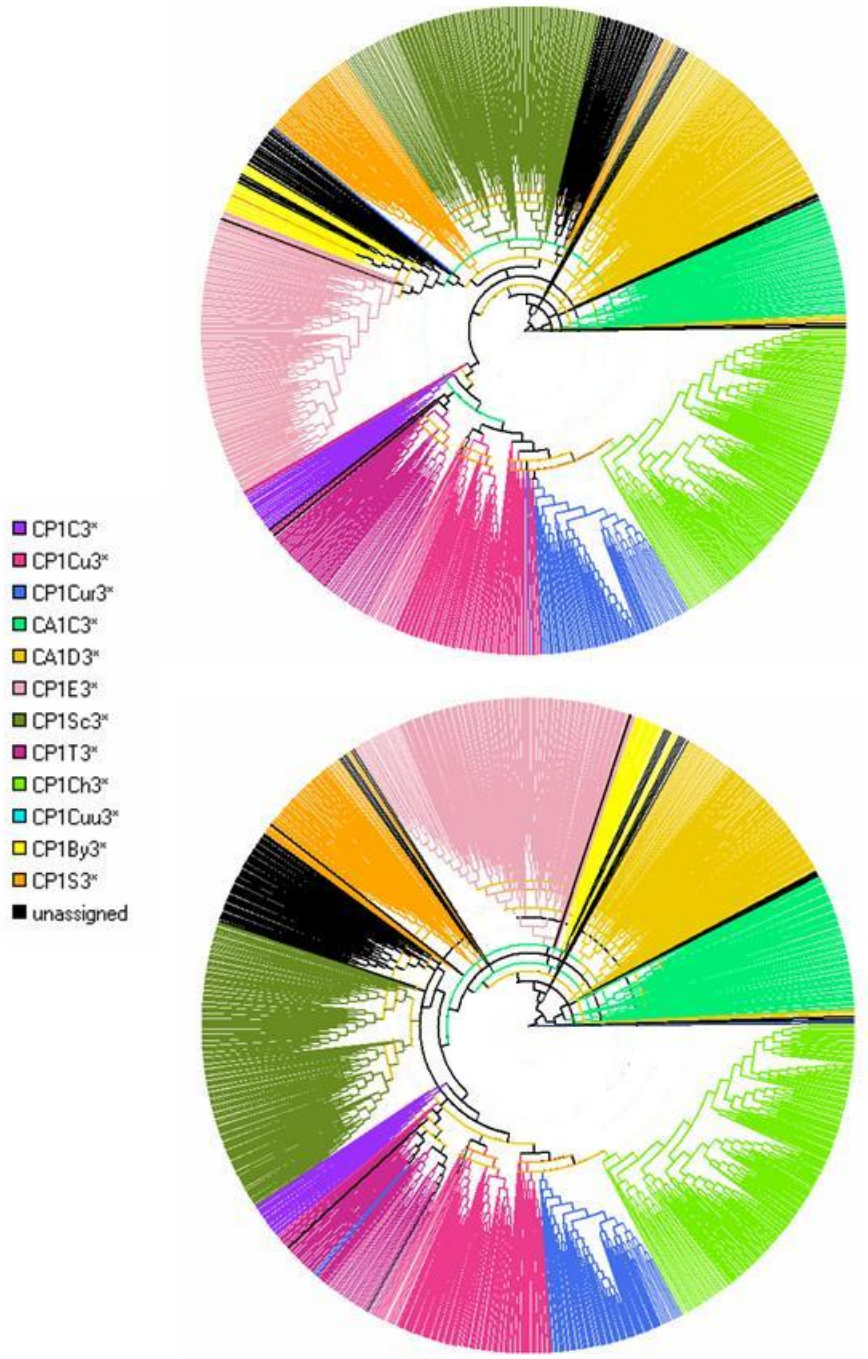


Figure 4.7: ML phylogeny of the Coleoptera built from the cluster-filtered (upper) and taxonomic filtered (lower) supermatrices. tRI = 0.872 for both trees. Trees are visualised using TreeDyn (Chevenet et al., 2006).

4.4: Discussion

There is great interest in the delineation of species boundaries using molecular data. In the current instance I adopt a basic clustering procedure for rapid, large scalable species delineation using unaligned loci. It is assumed this procedure is too inaccurate for general species delineation, and this appears to be indicated by the degree of incongruence with named species and between loci. For the current purpose incongruence was required in order to better demonstrate problems faced when concatenating sequence delineated loci.

Yet the procedure may prove useful in some instances, particularly, where unidentified data is present. Species assignment appears quite reliable in these cases, with the advantage that the clustering may easily be incorporated into the typical large-scale phyloinformatics pipeline. In the case of Coleoptera there are in the region of 1000 putative species which have not been identified to species level, for the cox1 gene alone. This amount of sequence diversity cannot be ignored. The assignment of species names to a proportion of this diversity would go some way to dealing with this particular limitation of mined data. A second species labelling issue apparent from the results presented here is that of misidentified/mislabelled sequences. Again, such a clustering step would flag candidate mislabelled sequences. Where taxonomic disparity between other members of a cluster, and the label, is great, mislabelling may be considered most likely. But in the cases where multiple closely-related species names are contained within a given cluster, it is conceivable that some cases of misidentification may exist here also. The cause of the large degree of incongruence is unknown, but intuitively, there is no reason to assume the cutoff for within/between species DNA sequence variation is constant, but one would expect lineage specific values due to variation in the rate of evolution, and variation in speciation times. It would be unwise to invoke a cause of misidentification when the extent of other confounding influences is unknown.

While the use of BlastClust for species delineation may be unadvisable, what of the adoption of a modified BlastClust delineation method, which accounts for uncertainty in species assignment? I have shown that, particularly for slow evolving loci, clustering of two species is often not share at other loci. One option in these cases would be

disregarding clustering in these cases. However many issues arose in attempting to implement this, with a great many arbitrary decisions required in creating new files with just agreement clusters. For example consider a cluster that contains two identified species. Were these not clustered at a second locus, this may prompt an attempt to split the cluster. However additional (perhaps unidentified) sequences in the cluster would then have an ambiguous species assignment. Further, when dealing with multiple loci, the likelihood of agreement clustering diminishes, so the fundamental principle of species status at a given threshold is dissolved. Still, even with agreement clusters, the issue of multiple possible concatenate combinations exists. So how would the methods presented here be improved? While the search may be limited to a high scoring region of the combination space, the search itself remains a random, unguided process. Given a high scoring combination, ideally an attempt would be made to improve on this combination before proceeding further. Given the practically permuted list of paths, this could be achieved by further permutations of subsets of the list of paths, perhaps attempting to improve the poorer sections of the combination. This would be akin to standard tree searching, whereby branch-swapping improves a number of random starting points. Where in the current case, the starting points are not random, but 'near'-optimal.

Even with such improvements, some fundamental limitations remain. Most importantly, where identified species are in separate clusters, the choice presented when concatenating to a member from a different locus presents a problem, as without further information (e.g. specimen labels), only a random concatenation can be made. Of course the benefit of the current method comes from how this random choice affects downstream choices (which influence the likelihood of finding a good concatenation combination). However the limitation remains, where one locus finds a taxonomic species as two putative sequence-based species, it is not possible to correctly assign which of these sequence-delineated species to a cluster from a second locus. Were representatives of both populations present in the second locus, one could validly assign to either, but again this requires labels (e.g. specimen tags, or population / geographic data) not typically available in the databases being used.

Analogous to gap penalties use in sequence alignment, the results of the optimal combination search are dependent on the scoring decisions. Although scoring of a whole

matrix favours completion (i.e. minimal missing data), so matrix size for optimal combinations are very similar. Scoring options implemented during the comparison of clusters from two loci are for a match (default = 2), specimen match (default = 1, this is in addition to the species match), and partial match (default = 1, partial match means a cluster contains both matches and mismatches when compared to a cluster from a second locus). Under the default scoring scheme, the concatenation of two cluster composed of identical specimen labels (and no other labels/ species) is favoured by the greatest score (three). This is a reasonable scheme as the sequencing for different loci has taken part on the same individual in these cases. Also giving score incrementation under default setting are the concatenation of clusters that contain different species, but only if there are species matches in addition to this. The choice of scores reflect the expectations of the user. For example he/she may wish to weight against partial matching if these are considered undesirable. If tip level relationships were under analysis (perhaps in a genus level phylogenetic study), allowing partial matching may be regarded as confounding. But where deep level nodes are of most interest, the creation of concatenates from different taxonomic species may be of no concern. Also it has the added advantage of matrix completion. Assuming the clustering of more distantly related taxa becomes increasing likely to be an erroneous outcome, it would be reasonable to place a taxonomic limit on the level of taxonomic allowed, perhaps by downweighting combinations that cross generic (and higher) boundaries. But the concept of assuming diversely labelled clusters are the same entity, invites more general discussion on what these clusters (and concatenates of) represent.

In the ideal situation, sequence-based species delimitation would take place after alignment concatenation. Specimen labels would allow the different loci sequence from individuals would be matched, permitting the creation of a supermatrix containing within species data. Species delimitation would use this supermatrix. For large-scale mined data, this approach is not feasible for a number of reasons, primarily the non-availability of specimen tags for most sequences make concatenation impossible, but further issues would be the computational demand of aligning non-filtered loci, and the lack of delimitation methods allowing between locus parameter variation. The individual locus species delimitation make incongruence more obvious, and it is this incongruence that we

are attempting to deal with here. But, do these incongruent clusters deserve 'putative species' status. Firstly, perhaps some loci are simply unsuited for these procedures. In the current analysis the greatest incongruence is found in the slowly evolving 18s and 28s loci (table 4). While they are clustered to maximise the return of taxonomic groups, it is presumed that the evolutionary characteristics (perhaps insufficient sequence divergence at the species level results in high variance) are not suited to an analysis at the species range. The result being a greater number of closely related species that are incorrectly grouped, and a greater number individual species (perhaps with relatively high sequence variation) erroneously split.

So, assuming analysis were limited to a subset of loci (e.g. cox1 and 16s perform relatively well for the Coleoptera), and there was a decision to concatenate a cluster with overlapping taxa, the issue remains: what are these clusters, are they valid mOTU's, and is their combination justified when other taxonomic species are present. The current method may be justified in these terms in that it tries to minimise the amount of taxonomic incongruence, and of course options may be set to disallow the pairing of clusters with some degree of taxonomic mismatch. But the assumption has been made that a cluster of sequences at one locus has been deemed to be an individual 'species'. This assumption would seem to have been disproven if a second locus contains only some taxonomic overlap. As such, where incongruence occurs, it may be incorrect to apply species delineation terminology to individually clustered loci, only to the concatenates. The relationship between individual locus clusters and the concatenated forms are similar to that between gene trees and species trees. Indeed, related methods may be applied in the creation of both mOTU's and gene trees. As such, it is entirely reasonable that supertree methods used in combining species trees be adapted for combining molecular-delineated clusters from different loci. Supertree methodology is well developed, and current procedures allow for incongruent signal across sources (Bininda-Emonds, 2004). The increased use of species delineation calls for the development of such source combination methods.

Chapter five: Site-specific rate variation in cox1 does not adversely impact its phylogenetic utility.

5.1: Introduction

Over recent years, cox1 has been a popular locus for evolutionary studies in insects. With the choice of cox1 as the barcoding marker, its relative presence in sequence databases will only increase. Its popularity is due to a number of convenient features. From a practical perspective, the primer sites are highly conserved, making PCR reliable across a broad range of taxa. Despite this conservation, the evolutionary rate is relatively fast, guaranteeing informative positions even in studies of speciation. Finally cox1 is largely regarded as length invariant, which avoids the pitfalls associated with multiple sequence alignment (Wong et al., 2008).

Discussion of the utility of cox1 in phylogenetic analysis has focussed on taxon sampling, since it is reasonable to assume that many taxonomic groups will be fully represented at the cox1 locus before long. This dense taxonomic sampling has led to the suggestion that cox1 may provide a convenient framework on which phylogenies may be constructed (Hajibabaei et al 2007), but the use of this locus as a framework requires a thorough understanding of its molecular evolutionary properties, in particular, those properties which violate the assumptions of commonly used phylogenetic methods.

Studies into the molecular evolution of cox1 in the insects have generally focused on two particular aspects: those of the pattern of variability across the locus, and of saturation. Early work was carried out by Lunt et al. (1996), who analysed an alignment of nine insect cox1 sequences. The variability was recorded as the average number of amino acids in a given region, after translation of the DNA sequences. The authors found significant difference in variability between COOH-terminal domain and the other structural classes (NH2 terminal, transmembrane, internal and external domain classes). There was no significant different between the other domain classes, but major difference between individual class members (e.g. the interal2 and internal4 domains showed great variability). Howland and Hewitt (1995) found similar variability patterns in the translated DNA sequences of 37 beetles. One of the few detailed molecular evolutionary analyses of cox1 in the beetles was carried out by Martinez-Navarro et al. (2005), using

the 3' end for 119 specimens of the Harpalini tribe. The pattern of translated amino-acid variability was found to be similar to that of earlier findings, but with a much greater variability at the COOH-terminus. The authors examined saturation at various taxonomic levels using different methods and found saturation at the sub-tribe/tribe level. In codon partitioned analyses the saturation is particularly marked at the 3rd position, along with a particularly high AT-bias. Yet this partition appears to be useful in that the resulting phylogeny is incoherent when it is removed. Similarly, Cognato and Sperling (2000) find the highest number of phylogenetically informative characters at this partition.

The non-standardised practice of using varying segments of cox1 prompted Roe and Sperling (2007) to study variability and saturation as a function of the size and location of the fragment (the study extended the analysis region into the adjacent cox2 locus, limiting the number of insect taxa suitable for study to 73). The authors find the barcode (5' cox1: between LCO/HCO primers) region to have the greatest within species diversity whereas the greatest between species diversity was at the 3' end of the gene (i.e. between the Jerry/Pat primers), although there was great variation in the results depending on the particular species pair analysed. A sliding window analysis of saturation and phylogenetics signal was performed, assuming the desire to both maximise divergence and minimise saturation. However no consistent patterns were found.

This variation of molecular rates along the string of a DNA sequence has been well studied (Yang, 1996a; Cunningham et al., 1998), and typically approximated by the gamma distribution in standard likelihood models (Yang, 1994). Models in general use do not allow for rate variation at individual sites, across the tree. However, statistical analyses of individual sites in sequence alignments have shown that rate change between subsections of a clade are commonplace (e.g. Lopez et al., 1999; Baele et al., 2006). One widely used method for detecting rate change at individual sites is the χ^2 test of homogeneity. If rate is constant throughout the tree, the number of substitutions occurring in a sub-section will be proportional to the size of the sub-section (i.e. the tree-length of each sub-group). The tree-lengths can therefore be used as expected counts for a χ^2 test at each site in the sequence, with significant deviation in observed substitution counts taken as evidence of rate change at the given site. As applied to the question of rate shifting, the basic χ^2 test has undergone recent improvements, for example making the test insensitive

to the length of the matrix (Baele et al., 2006). The finding that this form of rate change is common has prompted the development of evolutionary models that incorporate site-specific rate variability.

The covarion hypothesis is a specific form of SSRV which suggests that at a given time, only a proportion of sites in a protein are free to change, but the distribution of these variable sites changes over time (Fitch, 1971b). Thus the rate of a site in the covarion model is allowed to vary between two separate states; invariant, and the allocated variable rate. A number of generalisations of the strict covarion model have been developed. Tuffley and Steel (1998) describe a simple model containing a Markov switch, between site variability and invariance, with no further restrictions on switching applied. The covarion-like model of Galtier (2001) allows greater variation of rate change, as opposed to the discrete options of *variable* or *invariant*. Here, characters may evolve along branches according to rates, that have themselves evolved along the same branch. More recently, the model of Wu et al. (2008) contrasts with that of Galtier in that site rates are assigned independently to each edge in the tree, rather than shifting according to the 'rate of rate change' parameter. These various implementations of heterotachy often describe DNA sequence evolution more accurately than stationary rates-across-sites models (Galtier 2001; Zhou et al., 2007).

Research into the effects of heterotachy on phylogenetic inference is far from resolved. Simulations have clearly demonstrated that the presence of specific forms of heterotachy are capable of misleading tree-searches. Kolaczkowski and Thornton (2004) describe a biasing model in which sites are classed under two possible branch lengths, implying a shared rate change. Zhou et al. (2007) find little support for this particular form of heterotachy. Ruano-Rubio and Fares (2007) show that in situations where substitution rates are conserved in non-sister lineages and dissimilar in sister lineages, non-sister attraction occurs when a stationary gamma ML (maximum likelihood) model is applied. However it has been suggested that the acute forms of heterotachy simulated in these studies may be unlikely in empirical data (Wu et al., 2008), although the more generalised forms may be common. Further, the retention of heterotachous sites in SSU rRNA has been found to support uncontended eukaryotic monophyletic groups (Baele et al., 2006). It is necessary to assess more accurately the mode and result of heterotachy in

empirical data before the necessity of applying heterotachous models can be determined. At the same time, the phylogenetic utility of cox1 needs to be more accurately placed. To this end I examine SSRV in the Coleoptera. Following recent progress on the phylogenetics of this highly speciose order (Hunt et al., 2007), it is necessary to assess the influence SSRV is having on tree-searching. Further, as it is yet to be determined whether the structure of SSRV in empirical sequences is of the potentially biasing form discussed previously, I then analyse site rate change in terms of phylogenetic relatedness.

5.2: Methods

Data Collection and matrix formation

Coleoptera sequences were taken from the Genbank invertebrate release files (ftp://ftp.ncbi.nih.gov/genbank/) and searched using 31 Coleoptera 3' cox1 query sequences [make_datasets_with_blast.pl]. The blast output files were parsed using code from the BioPerl suite (Stajich et al.2002) and fasta format sequences retrieved utilizing the NCBI tools, fastacmd and formatdb (available from ftp://ftp.ncbi.nih.gov/toolbox/). The returned sequences were aligned with HMMalign (Eddy, 1998) under default parameters, then trimmed. I find a case of length variability in the alignment, despite cox1 being largely regarded as length invariable (it is common to see alignments composed of representatives from diverse orders and even phyla, with no internal gaps; Lunt et al., 1996; Saraste, 1990). The rate of amino acid substitution is noticeable higher in the region surrounding the amino indel. Species of the leaf beetle genus *Neochlamisus* contain single insertions at this position. The amino acid insertion maps to I4 domain, close to the I4 / M9 border. The sequences were filtered to leave a single entry per species, then well represented groups (genera with >18 sequences) identified for further analysis.

Subgroups were separated from the alignment then separate phylogenies were built for each, using the parsimony package 'Dnapars' (Phylip v3.67; Felsenstein, 1989). As Dnapars treats gaps as a fifth character state, these were recoded as missing data prior to running the analysis. Both the treelength and site-specific substitution counts were parsed from the Dnapars output file and read into R for statistical analysis (R Development Core Team, 2005).

Test of site rate heterogeneity

Heterogeneity in rates was tested at the matrix level and for individual sites. Firstly I test for a significant level of heterotachy in the dataset as a whole. The matrix was used as a two-way contingency table, with the expected number of substitutions for each position given by: (row_total * column_total) / number_of_observations. The χ^2 test statistic is then computed from 10000 Monte Carlo replicates (Hope, 1968). Next,

individual sites undergoing rate change were identified using the modified χ^2 test given in equation 2 of Baele et al. (2006). The expected count as used by Baele et al. factors in both the sum of all tree-lengths along with the total number of site substitutions. The advantage being that the method is not sensitive to factors such as the number of sites in the sequence (which can alter the tree length). In cases where tree lengths are very low, I multiply each tree-length by a constant in order to bring the lowest value above unity.

The χ^2 test is only recommended when the number of counts (substitutions) in each cell is >5. To overcome this limitation I use the permutation procedure of Roff and Bentzen (1989). The p-value is calculated by a comparison with a distribution of p-values produced from randomly reassigning the substitutions among the groups. To achieve this, the number of substitutions at a given homologous position are summed for the groups. Then each substitution is reallocated amongst the groups, with a probability proportional to the size of the group. A χ^2 test is repeated for each randomisation, with 5,000 permutations performed at each site, the final p-value for the site is calculated from the position of the unpermuted χ^2 value within the distribution of permuted values. Only positions undergoing a total number of substitutions which is greater than half the number of groups are considered testable (Lopez et al., 2002). As a significance test is performed for each testable site, standard cutoffs are not appropriate. I account for multiple testing using the method of Storey (2002), using the Q-value software. The package converts a list of p-values to q-values. By rejecting the null hypothesis only for sites with a q-value <0.05, I ensure an acceptable false discovery rate of 5%. Given the number of significantly heterotachous sites for each coding partition (1st, 2nd and 3rd), I test for an overrepresentation in the non-3rd partition by permuting the significant sites with respect to the partitions, with probabilities according to the number of testable sites in the respective partitions.

Simulation study

I estimate the false positive rate of the individual site χ^2 heterotachy test by a simulation study using a well studied dataset composed of four subdivided tiger beetle (Cicindelidae) clades (Vogler et al., 2005). Substitution parameters are inferred from the dataset then used to simulate homotachous sequences. The rate at which the null

hypothesis of homotachy is incorrectly rejected by the χ^2 heterotachy test on the simulated sequences gives an indication of the number of false positives when the χ^2 test is applied the current dataset. Sequences are simulated using parameters derived from ML searches of the combined tiger beetle sequences. The trees are inferred by constrained (with respect to the four clades) searches using RAxML (Stamatakis et al., 2005), applying the GTRGAMMAI model with partitioned 3rd and non-3rd data. Evolutionary parameters are then parsed from the RAxML output (base frequencies, alphas, invariant proportions, and GTR rates, for both the 3rd and non-3rd partitions) and used to simulate DNA sequence evolution. Sequences are simulated along the topology obtained from the ML search, with seq-gen (Rambaut and Grassly, 1997). A GTR model of evolution is applied, with rates assigned from a continuous gamma distribution. The simulated sequences are then separated into the four subclades and subject to the heterotachy analysis as previous.

Impact of rate heterogeneity on tree-searching

The influence of heterotachous positions on the tree topology is assessed by removal of the sites from the matrix prior to phylogenetic analysis. Phylogenetic trees were built under ML and parsimony optimality criteria. For ML I use the RAxML package, applying the GTRCAT model with 20 rate categories. To build parsimony trees I use TNT (Goloboff et al., 2003), with the default driven search and a 60 minute timeout (approximately the same running time for a single ML tree-search on the current dataset). The quality of the resulting phylogenies are measured by taxonomic congruence. The use of taxonomic congruence in assessing the quality of a locus, sequence alignment, or phylogeny has a long history (Wheeler, 1995; Zardoya and Meyer, 1996; Rokas et al., 2002). Here, I build on earlier work (Hunt and Vogler, 2008), to complete an automated tree assessment procedure that utilises the taxonomic data associated with each sequence. Each taxonomic grouping in the dataset is used as a character, with the number of changes between states calculated on the phylogeny generated previously. If the phylogeny is congruent with this character (i.e. the particular taxon is clustered in the tree), it will undergo fewer state changes than an incongruent phylogeny. The taxonomic RI (congruence score) can be calculated using each taxonomic grouping at a specific rank

or each taxonomic groupings at all ranks jointly (ensemble RI). The probability is obtained by comparison of the taxonomic congruence score when heterotachous sites are removed, to congruence scores obtained after removal of random homotachous positions.

Rate convergence test

Convergence in site rates amongst non-sister lineages following rate shifts has previously been shown to introduce topological bias (the 'bad' case given in Ruano-Rubio and Fares, 2007). Briefly, if the rate at a heterotachous sites changes as to result in a pattern whereby closely related (sister) clades show a reduced rate correlation over unrelated clades, the resulting character states will favour the clustering of the unrelated clades (Figure 5.1).

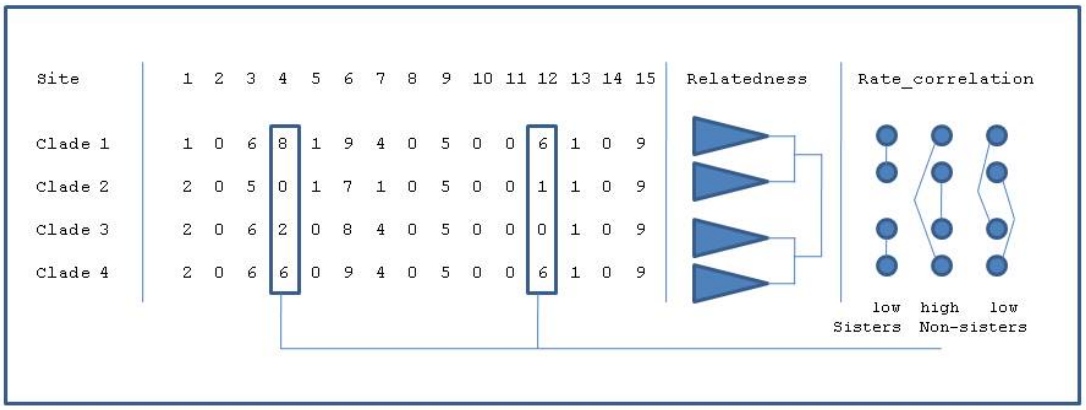


Figure 5.1. Illustrating the biasing form of heterotachy, and its relation to the traditional test of heterotachy. A sequence alignment of 15 sites is split into four clades (there are multiple sequences per clade), then the rates inferred for each clade (shown beneath site numbers). Heterotachous sites are identified via a χ^2 test, as those with a significantly different rate between clades (two sites are identified here and boxed). A tree-search may be biased towards the incorrect topology if the resulting pattern of rates correlate higher between non-sisters than sister clades. In this example, bias is introduced due to more similar rates between the non-sister pairs 2 & 3, and 1 & 4.

I tested for this pattern of rate change by comparing rates in sister groups to those in non-sister groups. To assign groups to sister pairs I take both the molecular phylogeny and taxonomic nomenclature into account. There is then a pairwise comparison of each sister pair to a subset of the other sister pairs. As susceptibility to bias is minimal when the sister pairs are separated by a large branch, I limit the comparisons to sister pairs with an adjoining branch-length of less than 0.08 substitutions. The length of the inner branch

length is calculated using the Paup distance function (Swofford, 2000). The topology was constrained with respect to the sister and non-sister assignments, then distances were estimated by ML with gamma distributed rates. For each comparison the product of the correlation (the R function used is: cor.test(rate_ranks1, rate_ranks2, alternative = "greater")) between the sisters is deducted to the product of the correlation between non-sisters, with two tests for the two possible non-sister conformations. The correlations are performed on the site substitution counts inferred in the heterotachy analysis, with only significantly heterotachous positions included. These sites are ranked in order to reduce sensitivity of the analysis to differences in treelength. Site substitutions typically vary between 0 and 10, with many ties expected. Ties were broken randomly, then the correlation value taken as a mean value after 40 repetitions. A p-value is calculated from a distribution of the same measure, generated by repetitions of the calculations using randomly chosen groups (four groups for each repetition). I assess various parameters for their ability to influence susceptibility to this form of bias. These parameters include adjoining branch length, alpha, subtree height, number of taxa, and ratio of tip to deep branches. In the case of the tip / inner branch ratio, I test for significant deviation in expected length of only the branches leading to the tips of the (unrooted) trees, by comparing the sum of these branches to the distribution after permutation (100 replicates) of branch length with respect to position on the tree.

5.3: Results

The Coleoptera cox1 dataset

The Blast search of the Coleoptera database gave 10,150 hits to the query sequences. After filtering, 3990 unique species id's remained. Many of these sequences were highly similar, with the BlastClust algorithm forming 480 clusters where sequence similarity was greater than 98. A single member of each of these clusters was taken, leaving a total of 3640 sequences. This is a substantial increase over previous molecular evolutionary studies, particularly of cox1.

After alignment and removal of truncated sequences, sequences for 29 densly sampled genera were retained. Twelve of these were sequences of the suborder Adephaga and 17 of the suborder Polyphaga. Groups sampling for the Polyphaga was varied, but the Dytiscid family was overrepresented in the Adephaga. Separate parsimony trees were inferred for each of the 29 genera, then site substitution counts parsed for analysis. The pattern of variability was first plotted by summing the substitution counts across the groups, then calculating mean values from a sliding window (Figure 5.2). As expected, the difference in variability across the partitions was marked, with the 3rd codon positions showing the greatest degree of variability and the 2nd codon positions showing the least. Also notable was the variation in variability across the cox1 axis, which was even apparent at the 3rd codon (despite this partition being largely silent).

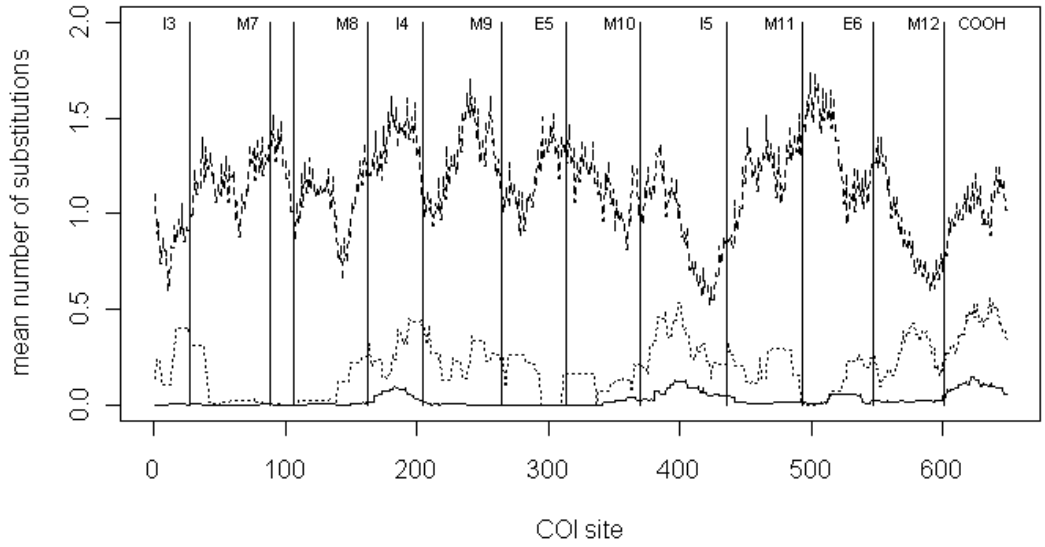


Figure 5.2. Sliding window plot of variability in the beetle cox1 nucleotide alignment. Variability is given as mean number of substitutions in a 24 base window. The three values are separated for the three codon positions, with the upper dashed line representing the 3rd partition, the mid dotted line the 1st partition, and the lower solid line the 2nd partition. Vertical line represent boarders between domains, with the domain identity of a region within two given boarders, specified in the upper region of the graph (I = internal, M=membrane, E=external). Values along the x axis correspond to position in the current cox1 alignment, not position along the cox1 gene.

A number of studies have analysed amino acid variability and found peaks in the I4, I5, M12 and COOH domains (Saraste, 1990; Lunt *et al*. 1996; Cognato and Sperling 2000; Martinez-Navarro *et al*. 2004). These results are mainly confirmed here. When partitioned according to codon position the pattern in amino acid variability most closely resembles that at the non-3rd codon positions. Conversely, the membrane spanning domains show a lower level of non-synonymous variation. The variation in substitutions has implications for the expected pattern in SSRV. The substitution threshold for SSRV significance testing mean the availability of testable sites will differ substantially both across the codon partitions and across the gene axis. For example there are virtually no testable sites in the first quarter of the cox1 fragment at the 2nd codon partition.

Rate heterogeneity

The matrix of cox1 substitution counts for the 29 groups was subject to contingency χ^2 of rate homogeneity test after removal of invariant columns. Significance was assessed for the whole table, and datasets composed of the 1st, 2nd and 3rd codon

partitions in isolation, showing highly significant deviation from homogeneity in all four cases ($p<0.001$). Given this evidence of heterogeneity in substation rate in the dataset, I then attempt to identify the heterotachous positions themselves. In line with the recommendations of Lopez et al. (2002) I tested for heterotachous behaviour at sites in which the total number of substitutions across the groups equated to greater than half the number of groups. In the current case, testable sites are regarded as those with greater than 15 substitutions, as summed across the 29 groups. The first and second positions had only 68 and 10 testable sites, respectively, while 218 of the 224 third codon positions could be tested for heterotachous behaviour. Typically the expected number of substitutions within each group are calculated from the tree-lengths of each group. Using this approach I find a marked difference between the largely silent 3rd codon position and the non-silent, non-3rd positions. Only 12% of the 3rd codon positions were found to be evolving under a heterotachous model, as opposed to 60% (41 out of 68 testable sites) and 70% (7/10) for the first and second codon sites, respectively. However, in cases where the chi squared test is applied to separate partitions it may be preferable to use partition-specific tree-lengths. The tree length is heavily weighted towards the 3rd codon partition due to its greatly increased substitution rate, meaning these values may not be reliable for use as expected rates in the non-3rd positions. This would be a severe problem if the 3rd and non-3rd treelengths were not correlated. For example, if a given group had a relatively fast rate (compared to other groups) at the 3rd codon without a correspondingly fast rate at the non-3rd (i.e. non-correlated), the fact that the overall tree-length generally represents the mean rate at the 3rd positions make it unsuitable as a measure of expected rate at the non-3rd sites. Although codon specific tree-lengths are correlated (between the 3rd and 2nd tree-lengths: $r=0.42$, $p=0.02$, Pearson's product-moment correlation), the 3rd codon tree-lengths show much less variation across the groups than the non-3rd ($p<0.001$, F test of variances). With this in consideration, the tests were repeated using tree-lengths calculated separately for the 3 codon positions. This resulted in a drastic change at the non-3rd partition. Similar to the previous analysis, 10% (22/224) of the 3rd positions were heterotachous, but the 1st at 2nd proportions were reduced to 47% (32/68) and 10%, (1/10) respectively. Although both the tests using overall tree lengths and those with codon

specific tree length showed a significant overrepresentation of heterotachous sites at the non-3rd partition ($p<0.001$ in both cases).

The false positive rate of the χ^2 test heterotachy test was assessed by simulating datasets composed of entirely homotachous sites, on which the test was applied. The parameters used for the simulation were obtained from ML analysis of a previous Coleoptera dataset. 50 tree-searches were performed on the dataset, and the parameters obtained (table 5.1) in each replicate were used for simulation. I find very few of the simulated homotachous sites were incorrectly identified as heterotachous. The proportion of testable sites with a significant q-value was <0.01 in all simulation replicates (mean 0.0046 sd 0.00248). This is not expected to impact the 296 testable sites in the current dataset.

Partition	Alpha	Base Frequencies				GTR Rates						Inv
		A	C	G	T	A-C	A-G	A-T	C-G	C-T	G-T	
1/2	0.488	0.291	0.157	0.174	0.377	0.83	10.60	2.06	0.58	13.08	1	0.558
3	0.126	0.389	0.160	0.074	0.378	6.19	101.57	5.91	4.13	76.32	1	0.017

Table 5.1: Mean values retrieved from ML analysis of the Cicindela dataset, and used during simulation of homotachous sequences.

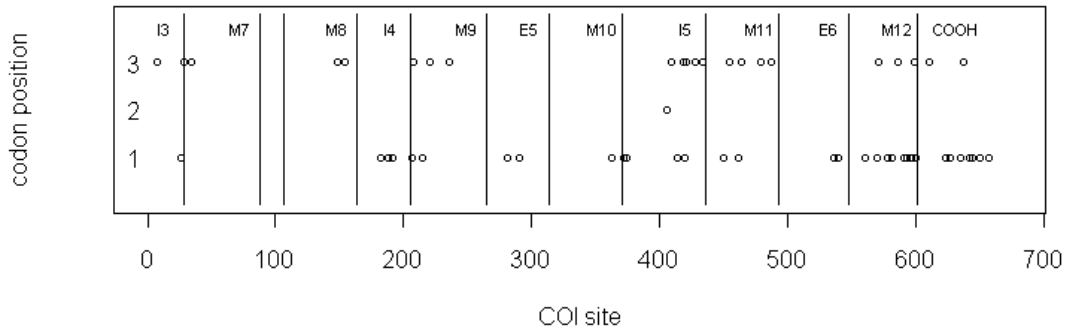


Fig. 5.3 Significantly heterotachous sites in the Coleoptera cox1 dataset. The heterotachous positions in the 1st, 2nd and 3rd codon partitions are shown in the lower, middle and upper row of plotted points, respectively. Domains are indicated as in figure 5.2.

The heterotachous positions were mapped on to the domain structure of cox1 (Fig. 5.3). The M12 membrane domain and the COOH domain show the greatest number

of heterotachous positions, which is likely due to the increased variability at the C terminus (which in turn allows a greater proportion of sites to be tested for heterotachy).

Impact on tree-searching

Given the bias that SSRV may assert when using rate-stationary models, it is necessary to assess the effect that sites undergoing this mode of substitution have on the topology. Trees were built from the alignments using RAxML and TNT, then the taxonomic framework used as an independent (morphological-based) standard by which to assess topological accuracy. The phylogeny built from the alignment with all sites included gave an ensemble taxonomic RI of 0.972, and genus level and family level RI's of 0.939 and 0.997, respectively (table 2). When the 55 heterotachous sites are removed there is a reduction in RI to 0.957 (0.902 at genus level and 0.991 at family level) (table 5.2).

	All_sites_included	Heterotachous_removed	Homotachous_removed (random)	Homotachous_removed (rate_standardised)
Parsimony				
Ensemble	0.972	0.957	0.967	0.967
Genus	0.939	0.902	0.925	0.929
Family	0.997	0.911	0.991	0.990
ML				
Ensemble	0.983	0.974	0.978	0.978
Genus	0.948	0.935	0.939	0.942
Family	0.999	0.966	0.977	0.977

Table 5.2: Taxonomic RI scores calculated from trees inferred from the cox1 alignment. Trees were built using Parsimony and ML, then RI calculated using all ranks (ensemble), at genus level only, and family level only. Single values are obtained for the full alignment (column 2), and for the alignment with heterotachous sites removed (column 3). Multiple replicates were performed for removal of homotachous sites, so mean values are given (columns 4 and 5).

To generate a null distribution I removed 55 homotachous sites from the dataset at random. After 100 repetitions the associated ensemble taxonomic RI scores were found to be normally distributed, with the Shapiro-Wilk normality test p-values > 0.05 in both cases (Royston 1982). Therefore I calculate significance via the area under the normal curve. The RI of the tree built after removing the heterotachous positions was found to be significantly reduced in comparison to the distribution of scores after removal of random homotachous sites ($p=<0.001$). It has previously been suggested that the reduction in

phylogenetic performance after removal of heterotachous positions is due to the fact that positions prone to this mode of evolution also evolve at an absolute rate that is particularly informative. I account for this with a second test which generated the null distribution by removing sites with a mean rate similar to that of the heterotachous sites. Yet alignments which have had heterotachous sites removed are still found to generate lower scoring trees than alignments with homotachous (and rate-normalised) sites removed ($p=<0.001$), with no difference in the RI mean values between removal of random sites and sites with the same variability scores (t test, $p=0.98$). It should be noted that the contribution to the change in RI as heterotachous sites are removed, differs across the taxonomic levels. RI differs very little at the higher ranks (e.g. family), whereas there is a substantial reduction at the genus level.

Comparison of ML and parsimony

There is continuing interest in the differing susceptibilities of parsimony and ML methods to heterogeneous substitution processes, including SSRV (Farris, 1999; Gaucher and Miyamoto, 2005). While optimality scores are consistently higher for ML trees, I see the same changes in score according to presence / absence of heterotachous positions. The taxonomic RI for the complete data matrix was 0.983 for the ML tree, compared to 0.972 for the parsimony tree. The ML trees undergo a reduction from 0.983 to 0.974 with the removal of heterotachous sites from the matrix. In the case of ML there is a modest difference between the means of trees built using alignments with random sites removed, or sites with rates similar to heterotachous sites (normalised > random, $p=0.012$ t-test). Although like the parsimony trees, both distributions are significantly higher than trees built using alignments with heterotachous sites removed ($p=0.001$ in both cases). The results indicate that the heterotachous positions are having a beneficial influence on the tree-search relative to the homotachous sites, irrespective of whether a parsimony or ML algorithm is used.

Test for biasing rate pattern

While the presence of heterotachous sites appears beneficial on the whole, the potential for heterotachy-based bias in subsets of the dataset exists. Bias may be introduced, not by heterotachy *per se*, but when rate change is structured in a particular pattern (Ruano-Rubio and Fares, 2007). More specifically, when rates do not change in accordance with the degree of relatedness. If this asymmetric rate change leads to unrelated clades having the most similar rates, the resulting character distributions lead to a clustering of the unrelated clades under rate stationary models. The question of whether rates change in such an asymmetric manner remain pertinent, as the analysis of heterotachy in empirical sequences is typically of a more generalised form. To test for the presence of this form of rate patterning, I compared the correlation in site rates amongst sister clades to that between non-sister clades. The authors give the change in the level of bias according to a number of parameters. Parameters in the current study that would be expected to increase susceptibility to bias would be the low alpha estimates and the sub-tree heights (the mean treelength for subclades was 1.3). Whereas the relatively high number of taxa per group (typically >20), the excessive tip level branch length (in all 29 groups except Illybus and Deronectes(?), table x), and no doubt substantial inner branch lengths (all sister pairs are compared irrespective of adjoining length) would decrease bias susceptibility. Thirteen reliable sister pairs were assigned sister status based on higher taxonomic groupings. Site rates at heterotachous positions were correlated both between sister pairs and non-sister pairs then significance calculated from 5000 permutations.

A distribution of p-values was obtained from multiple tests for bias for various sister combinations (Figure 5.4). Low values (<~0.05) represent cases where the site rate correlations are much higher between sister clades than non-sister clades, whereas high values (>~0.95) represent cases of increased correlation between non-sister pairs. The tendency is clearly towards higher correlations between sister rates, with no examples of the potentially biasing case of a greater correlation in site rates between non-sister clades than sister clades.

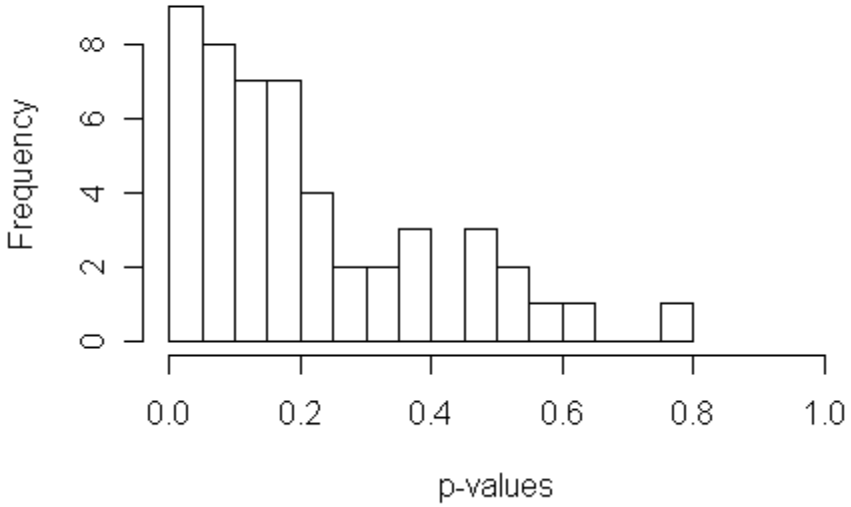


Figure 5.4 frequency histogram of p-values obtained from correlating heterotachous site rate between sister clades versus non-sister clades.

5.4: Discussion

Despite the concerning results of theoretic and simulation studies, my empirical analyses appear to suggest the presence of heterotachy are not having an adverse effect on tree-searching. Based on previous findings, one explanation is that heterotachy may be prevalent at absolute rates which are also phylogenetically informative (Baele et al., 2006). The removal of sites undergoing rate shifts results in a rate-homogeneous dataset, a dataset that conforms to the assumptions of rate-stationary models. But this improvement is swamped by the corresponding loss of phylogenetic signal. The site removal analysis described here has similarity to that performed by Baele et al. (2006), except an optimality score is used which considers all taxonomic nodes as well as the score limited to the key monophyletic groups. Our findings conflict in that the benefit of heterotachous positions are not fully accounted for by an informative absolute rate. Nevertheless, on balance there appears to be a benefit to retaining these sites, with only modest empirical evidence to the contrary (Cheng et al., 2008).

Our understanding of SSRV in terms of the pattern of variability across a tree is generally limited. It has been found that on some occasions different subsections of a tree may differ in their contribution to SSRV. Baele et al. (2006) found many sites at which individual clades in a multiply partitioned tree are responsible for heterotachy. At the opposite extreme, the contribution of a given clade to heterotachy would be a function of evolutionary distance, such that large rate changes tend to occur where large branches separate subsections of the tree. In other words, rate change would be restricted at adjacent braches, meaning that large rate change requires gradual transition between intermediate rates, and so is only possible over large evolutionary distance. This is analogous to the correlation parameter sometimes applied to among site variation model, where rates are more accurately described when rate change is restricted at adjacent sites (Yang, 1995). Hence, as well determining whether a site is constant or variable throughout the tree, an extra dimension exists in the manner in which site specific rate change occurs. In the current analysis I find many sites with this restricted rate change in closely related subclades.

It is reasonable to hypothesise that this factor may in part account for the positive influence of positions with a significantly heterogeneous rate. That is, while sites in the dataset may differ significantly in rate across the groups, the pattern in rate change does not occur in a manner expected to mislead tree-searches. On the contrary, the pattern of rate variability is structured in such a way as can only increase the likelihood of obtaining correct topologies. Simulation studies have been quite illuminating in terms of identifying more precisely the pattern of rate change in a sequence that results in characters that tend to mislead phylogenetic inference procedures. It is not simple shifts in site rate that mislead, but concerted site rate shifts in unrelated sub-clades, which reduce the correlation in rates amongst closely related groups (Ruano-Rubio and Fares, 2007). This form of heterotachy is easily simulated but may appear to be an unlikely occurrence in nature. Here I not only confirm the absence of this type of heterotachy in cox1, but find evidence of the opposing case, where close relatives show a higher correlation in rates in comparison to unrelated clades. It is unclear at this point whether a tree-search is indifferent to this pattern of variation, or whether the pattern is positively biasing. The latter instance may be indicated from the simulations discussed previously.

In the current case it is difficult to prove that sister correlation is behind the increased phylogenetic utility of heterotachous positions. Other hypotheses which may not been ruled out may include, for example, the robust clade definition given by the sub-clade invariance of heterotachous positions. Inspection of heterotachous sites (as assessed by the current method) leads to the observation that for the large part, they shift in rate to invariance in many of the sub-clades. This character state fixation will make these sites particularly informative for defining the clades. And as the monophyly of these same clades are typically the benchmark at which a tree is assessed, the inclusion of these sites increases the optimality score. Lockhart et al. (1998) found much of the support for the split between photosynthetic eukaryotes and prokaryotes due to sites that had undergone a shift from invariant to variable. A related class of sites have been called 'constant but different' (CBD; Gribaldo et al., 2003); these are sites which have undergone a character state change between subclades, but remain invariant thereafter. Presumably CBD sites provide substantial support for splits, but in practice they number too few for statistical

testing. If the support for heterotachous positions is due to subclade invariance, we might expect their behaviour to reflect that of CBD sites.

The results presented here suggest careful implementation of the χ^2 heterotachy test is required for protein coding sequences. It has been noted previously that the non-3rd (1st and 2nd) codon positions are enriched for SSRV (Ane et al., 2005; Vogler et al., 2005). As the non-3rd partition is exposed to much greater selection than the 3rd, one interpretation could be that changes in rate precede changes in selection pressure on individual sites. Our results appeared to confirm the increased SSRV at the non-3rd sites, but this conclusion was dependent on the way the statistical significance was calculated. Reducing the overweighting of the 3rd codon in the χ^2 expected counts (treelengths) resulted in more modest patterns across the partitions. Further, the absence of any positively selected sites in Coleoptera cox1 (data not shown) suggests no substantial rate change in the coding partition.

The level of heterotachy in Coleopteran cox1 is consistent from findings at other loci (e.g. Lopez et al., 2002, Baele et al., 2006). Interestingly, the heterotachous nature of cox1 in the Coleoptera has not prevented clock-like behaviour at this locus, in insects as a whole. Gaunt and Miles (2002) found global clock-like behaviour (particularly at the 2nd codon position) in insect cox1 (although the Coleoptera were not included in their study). Assuming the Coleoptera do not show a radical departure in clock-like behaviour from the insect subgroups used in the study, this raises the question of the influence that heterotachy has on clock-like genes. The standard test of a global clock is the likelihood ratio test, comparing a clock-constrained and non-clock-constrained model. However SSRV is absent in the widely used models of sequence evolution used to implement the global clock test. This means it is unlikely that the evolution of a gene with heterotachous sites is being accurately described in both models. Nor does a relaxed clock model account for SSRV, as change in the substitution parameter amongst branches in a relaxed clock are applied to all sites equally. Intuitively, heterotachous sites would be detrimental for molecular dating. Yet another form of SSRV may be more relevant here; rates may differ across the depths of a phylogeny, yet be uniform across within discrete ranges. That is, a site may experience a shared rate at the genus level, but with uniform rate shifts in the deeper branches. It could be reasonably hypothesised that the presence of such sites

would contribute to sensitivity of molecular dating to the depth of calibration (Ho et al., 2005). Testing this hypothesis awaits the development of models incorporating this mode of SSRV.

Chapter Six: General Discussion and Conclusions

6.1: The Analysis Framework; HIP-DB

The analyses in this thesis were made possible from the development of the HIP-DB system (summarised in Figure 3.1). The issues that need to be overcome when developing a phylogenetics pipeline, and the different ways to deal with these issue (implemented in HIP-DB or in a contemporary system), are specific to research questions, and so such pipelines always have unique features. Once on a local system, obtained sequences (usually a taxonomic subset of those retrieved from online repositories) need to be consolidated. One approach to consolidation is the similarity-based clustering of the original database, to create separate files of homologous sequences (e.g. McMahon and Sanderson, 2006; Sanderson et al., 2008; Thomson and Shaffer, 2010). This requires no prior specification of the genes to be used, which makes the system more adaptable to other taxonomic groups, but the clustering is computationally intensive, a problem for larger datasets. Where a bespoke system were applied to specific section of the tree of life, the prior knowledge (of the identity and suitability of well sampled loci) favours extraction of specific loci, usually by homology search (Jones and Blaxter, 2006; Golobof et al., 2009). This is the approach used in the current instance, as analyses are dominated by the Holometabola (or subset of). However, this creates of a small number of large files (containing the homologs). The subsequent alignment steps can be difficult due to the presence of quite divergent taxa. There are various ways of reducing the computational demand of large scale alignment. For example, avoiding the repeated intensive alignments by maintaining curated core/profile alignments, to which new sequence can be are aligned (Kauff et al., 2007; Wu et al., 2008), or separately aligning slices of the dataset (McMahon and Sanderson, 2006; Golobof, et al. 2009). Where the system allows the user to experiment with many different alignment programs (as with HIP-DB), this can be impractical to implement, instead I make available methods able to cope with a range of dataset sizes (Belshaw and Katzourakis, 2005; Katoh and Toh, 2008; Notredame, 2007). It is advisable to equip the user with the greatest possible control over alignment, given the sensitivity of results to the approach of this particular step (Wong, 2008; Borrow, 2010). Creating usable

supermatrices from these raw locus data files can in principle be easily achieved by taxonomic species filtering and separate alignment. Uniquely for HIP-DB, I implement options for either this traditional morphospecies approach, but also sequence based species filtering options. These include the method as described in chapter four (section 4.1), and also a hybrid method whereby sequence clustering is performed in order to assign species status to unidentified sequences, followed by the usual name based filtering. The incongruence's generated via sequence filtering are problematic for the concatenation step (an issue dealt with in chapter four). But few alternative approaches to combining sequence-delineated datasets exist. The approach of O'Meara (2010), combines gene trees with simultaneous species delimitation. This supertree-like method would be unsuitable for large-scale mined data in its current form. Integrating into a supermatrix-based species delimitation pipeline, I delineate prior to alignment, reducing alignment complexity. However delineation methods are not fomalised in this way (I simply co-opt a clustering algorithm), and so the field is wide open. The alternative delineation of a non-filtered but aligned supermatrix would pose sequence matching issues (see chapter four discussion). But this would be quite dependent on the dataset to hand.

Also unique to HIP-DB was the assignment of geographic zones to the trees, using data submitted to geographic web sites. The lack of formalised web services at these sites necessitated the use of scraping tools (Lester, 2005). Such scraping is not ideal. Regular maintenance of the scripts is necessary to prevent minor changes in the format of the website from deactivating the processes. The development of web services for these sites, and future integration of such geographic databases with Genbank, would facilitate large-scale automated geographic assignment, greatly expanding the scale of phylogeographic research.

6.2: Resolving the Tree of Life With cox1?

When using a system such as HIP-DB, it is obvious that cox1 will be dominant in many parts of matrix from which the tree of life is formed (compare the number of species represented by cox1, to other loci, in table 4.2). It is pertinent to ask in what way the matrix of life will evolve, and how will its structure influence the resulting phylogenetic

inferences. It has been well established that the commonly used loci differ in the characteristics which influence phylogenetic accuracy (Danforth et al., 2005). Mt genes show fast rates of evolution, saturation (Martinez-Navarro et al., 2005), and heterogeneous patterns of sequence evolution (chapter five). It would be reasonably assumed that these features would be detrimental to the ability to resolve deep internal phylogenetic nodes. But I find that where taxonomic overlap is strong (table 3.3, cox1 targeted rows), a fast evolving mt locus will still make significant improvements to an existing supermatrix, even at deeper taxonomic nodes. How might this be squared against the general feeling that cox1 should not be used for resolving nodes beyond genus level? Sperling (2003) reviewed a number of Lepidoptera studies and found some phylogenetic signal in mtDNA at subfamily level, although reduced in comparison to genus. Danforth et al. (2005) suggested insect mt genes suffered prohibitive levels of homoplasy above the levels of genus/tribe. Nazari et al. (2007) found relative poor performance of mt genes at deep levels compared to nuclear loci. Wilson (2010) found phylogenetic information content in cox1 was comparable to other tested loci at genus level, but reduced relative to them at higher taxonomic ranks. Homoplasy is prevalent where substitution rate is rapid, but in terms of node reconstruction, the pattern of variability believed to result in lack of deep level resolution is insufficient character change at non-synonymous positions, with saturation (and thus lost information) at synonymous sites. Although this is likely to be a simplification, as cox1 shows variation in synonymous and non-synonymous rates across the domains (Spicer, 1995), and across the gene axis as a whole (Roe and Sperling, 2007), meaning saturation is not expected along the whole axis.

Of critical importance in terms of cox1, the level of sampling density will likely negate these saturation issues (Hillis, 1996). With increased sampling density, long branches are transected and so previously hidden substitutions can then be correctly reconstructed (Hillis, 1998; Graybeal, 1998; Yang, 1998). Consistent with this, the increased sampling of cox1 can lead to node resolution even at deep taxonomic levels. Thus it may be the case that signal is reduced, in this thesis I have shown that phylogenetic information is still present, and the improvement is significant. But it is important to note that this perceived reduction in phylogenetic usefulness is relative to other markers. For example cox1 may not improve a phylogeny to the degree as a nuclear

rRNA gene (Figure 3.6), this does not stop a significant improvement in resolution when including cox1 in a datamatrix.

But I have demonstrated that the utility itself is very much dependent on a number of factors. Firstly, taxonomic structuring of the dataset. In order to extrapolate wider significance to these finding, it is informative to speculate on the way in which the matrix of life will fill out. Currently sequencing effort are drastically clustered (e.g. Table 3.1). The path towards the goal of a homologous sequence for each species would not necessarily immediately address this extreme clustering, as even densely sampled clades only contain a small proportion of the available diversity. We see from chapter two that where sampling is discretised, taxonomic monophyly is more likely. A similar effect has been found by other authors at the species level, resulting in artificially clean species boundaries (Wiemers and Fiedler, 2007). As sampling tends towards symmetry across taxa, we expect these 'artificial monophylys' to be removed.

As I demonstrate, the sampling effect is complicated by taxonomic overlap (chapter three). The likely increasing lack of overlap in barcode sequence with existing datasets may limit their impact on deep level nodes. Previous authors have analysed overlap in terms of the degree of missing data. The simulation study of Wiens (2003) suggests a lower limit are required, rather than a specified proportion of completeness. Concatenating a barcode dataset adds ~700 characters, which would appear to satisfy the results in Figure 2 of Wiens (2003), which shows little reduction in accuracy where >= ~400 characters are used. In the current study (chapter 3) I do not assess the accuracy of the additional taxa that are introduced upon inclusion of cox1, only their effect on existing taxa. But as sufficient characters are being introduced, it is likely these are reliably placed in the phylogeny. These earlier simulation results had let to the suggestion that a datamatrix may be built around a scaffold of taxa with many genes sequences, with the remaining (majority) of taxa having few (or single) loci present (Wiens et al., 2005). In terms of the matrix of life, cox1 barcode data may be added to any existing scaffold of taxa with greater character presence. The correct placement of additional taxa is perhaps of second importance to the improvement of the scaffold matrix. Here we see that the matrix sometimes undergoes an improved ability to recover the expected nodes, but with no evidence of worsening.

In chapter four I address difficulties in utilising multiple sequence delineated loci, when data is mined. Under the growing cox1 database, and the desire to attain an accurate tree of life, in which circumstances would these procedures be useful, is it likely sequence filtering will be adopted for such datasets, and if not, what alternatives may be of greater practicality or have more biological meaning. As discussed earlier, unidentified sequences may be easily dealt with in this manner. Thus as the cox1 locus approaches dense species completion, the unidentified sequence diversity may be more reliably delineated according to these references. But of course using morphospecies assignments is simply a barcode like approach, would a dataset of this type benefit from resetting species boundaries in all sequences (including those with a morphospecies label). The arguments relating to gene trees are particularly relevant here. The barcode efforts mean cox1 will gradually predominate relative to other loci. A single locus will probably show many incongruence's with the species tree, and so species delineation will be inaccurate (whether or not rough and ready clustering methods are used). Where thorough taxonomic treatment is not feasible, this may be acceptable, in which cases these dataset may be clustered such, then combined with other data. Where the supermatrix approach is used this would necessitate a combination method such as that introduced here.

6.3: Limitations; Assessing Tree Quality

Taxonomic consistency has been used extensively here, but it is important to address limitations in the current scoring system. The advantage of scoring trees by tRI come from both the perspective of the scientific method, and simple bioinformatic feasibility. The score is easily implemented (bioinformatic feasibility) as the groupings are implicit in the species labels attached to DNA sequences. The former requires a hypothesis be tested against an answer that has been arrived at independently. The Linnaean taxonomic system (as an ideal, given current thinking) represents independent evolutionary groups, inferred mostly by cladistic analysis of morphological traits. Therefore these morphological-based groups represent an independent estimate of relatedness against which molecular findings can be scaled. But science is an ongoing process, and the taxonomic system is not likely to a) represent the true evolutionary groupings b) represent morphological groupings. The former is unknowable, and only simulation

studies have the advantage of knowing the true tree. But the latter represents incomplete work on the current taxonomy. In cases where a given tree is scored against the Dynastinae node for example, a worsening of taxonomic congruence would not necessarily mean reduced accuracy, as correct polyphylys may be resolved. In the case of Dynastinae, taxonomic revision may not occur for years to come. So what course of action is possible in such cases? Taking the bioinformatic approach, this would usually just be an accepted source of error, as morphotaxa are congruent on the whole (Jablonski and Finarelli, 2009). The degree or error this represents has not been estimated, but it's a reasonable assumption that most major taxonomic groups are supported by multiple synapomorphies. For restricted, in depth analyses of specific taxa (where there is more knowledge of the subgroups under study), a key subset of taxonomic nodes may be scored for recovery. Yet this is somewhat unsatisfactory as may be difficult to avoid investigator bias, it is up to the worker to determine whether the morphological evidence for monopoly of each group, is sufficient. Thus it would seem that the current approach has the advantage of objectivity

6.4: Conclusions

The unrivalled taxon sampling of the cox1 locus make it a high priority for studies of phylogenetics and molecular evolution. The phylogenetic potential of cox1 needs evaluating, as it could play a substantial role in the construction of the tree of life. The bioinformatic approach has made feasible a thorough examination of the phylogenetic impact and characteristics of cox1. There appears to be a robust phylogenetic signal, with the ability to successfully recover clades even at deep (family) levels. This is despite the presence of molecular evolutionary features typically viewed as undesirable in phylogenetics, including saturation and SSRV. Taking this into consideration, cox1 could be an informative marker for large scale phylogenetic analysis, particularly in combination with more slowly evolving loci, but supermatrix completion is an important factor to consider when including this locus.

References

Ane C, Burleigh G, McMahon MM, Sanderson MJ (2005). Covarion structure in plastid genome evolution: a new statistical test. Mol. Biol. Evol. 22:914-924.

Avise JC, Arnold J, Ball RM, Bermingham E, Lamb T, Neigel JE, Reeb CA, Saunders NC (1987). Intraspecific phylogeography: the mitochondrial DNA bridge between population genetics and systematics. Annu. Rev. Ecol. Syst. 18, 489–522.

Baele G, Raes J, Van de Peer Y, Vansteelandt S (2006). An improved statistical method for detecting heterotachy in nucleotide sequences. Mol. Biol. Evol. 2 3:1397-1405.

Ballard JWO and Whitlock MC (2004). The incomplete natural history of mitochondria. Mol. Ecol. 13, 729–744.

Bapteste E, Brinkmann H, Lee JA, Moore DV, Sensen CV, Gordon P, Duruflé L, Gaasterland T, Lopez P, Müller M, and Philippe H, (2002). The analysis of 100 genes supports the grouping of three highly divergent amoebae: Dictyostelium, Entamoeba, and Mastigamoeba. Proc. Natl Acad. Sci. USA 99, 1414–1419.

Belshaw R, Fitton MG, Herniou E, Gimeno C. and Quicke, DLJ (1998). A phylogenetic reconstruction of the Ichneumonoidea (Hymenoptera) based on the D2 variable region of 28S ribosomal DNA. Syst. Entom. 23:109-123.

Belshaw R, Dowton M, Quicke DLJ and Austin AD (2000). Estimating ancestral geographic distributions: a Gondwanan origin for the aphid parasitoids. Proc. Roy. Soc. B, 267: 491-496.

Belshaw R, Katzourakis A. (2005). BlastAlign: a program that uses blast to align problematic nucleotide sequences. Bioinformatics. 21:122-123.

Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J and Wheeler (2004). GenBank: update. Nucl. Acids Res. 32 (suppl 1): D23-D26.

Beutel RG and Pohl H (2006). Endopterygote systematics – where do we stand and what is the goal (Hexapoda, Arthropoda)? Syst. Entom. 31, 202–219.

Bininda-Emonds ORP (2004). The evolution of supertrees. TREE 19(6):315-322.

Bininda-Emonds ORP (2005). TransAlign: using amino acids to facilitate the multiple alignment of protein-coding DNA sequences. BMC Bioinf. 6:156.

Borrow C (2010). PhD thesis. Imperial College London, Division of Biology.

Brock PM, Döring H and Bidartondo MI (2009). How to know unknown fungi: the role of a herbarium. New Phyt. 181: 719–724.

Brower AVZ (1999). Delimitation of phylogenetic species with DNA sequences: A critique of Davis and Nixon's population aggregation analysis. Syst. Biol. 48:199-213.

Brummitt R (2001). World Geographic scheme for recording plant distributions. Hunt institute for botanical documentation, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

Burleigh JG and Mathews S (2007). Assessing among-locus variation in the inference of seed plant phylogeny. Int. J. Plant Sci. 168(2):111–124.

Caterino MS, Cho S and Sperling FAH, (2000). The current state of insect molecular systematics: A thriving tower of Babel. Annu. Rev. Entomol. 45, 1–54.

Chevenet F, Brun C, Banuls AL, Jacq B and Christen R (2006). TreeDyn: towards dynamic graphics and annotations for analyses of trees. BMC Bioinformatics, 7:439.

Cognato A and Sperling F (2000). Phylogeny of Ips DeGeer species (Coleoptera: Scolytidae) inferred from mitochondrial Cytochrome oxidase I DNA sequence. Mol Phylog Evol 14:445-460.

Cunningham C, Zhu H and Hillis DM (1998). Best-fit maximum-likelihood models for phylogenetic inference: empirical tests with known phylogenies. Evol 52:978-987.

Danforth BN, Lin C and Fang J (2005). How do insect nuclear ribosomal genes compare to protein-coding genes in phylogenetic utility and nucleotide substitution patterns? Syst. Entom. 30, 549–562.

Davis JI and Nixon KC (1992). Populations, genetic variation, and the delimitation of phylogenetic species. Syst. Biol. 41:421–435.

Dayhoff MO (1979). Atlas of protein sequence and structure. Vol. 5, Suppl. 3, 1978. National Biomedical Research Foundation, Washington, D.C.

de Queiroz K (2007). Species Concepts and Species Delimitation. Syst. Biol. 56(6):879–886, 2007.

de Queiroz A and Gatesy J (2006). The supermatrix approach to systematics TREE 22(1):34-41.

Dereeper A, Guignon V, Blanc G, Audic S, Buffet S, Chevenet F, Dufayard J, Guindon S, Lefort V, Lescot M, Claverie JM, Gascuel O (2008). Phylogeny.fr: robust phylogenetic analysis for the non-specialist . Nucl. Acids Res. 36(Web Server issue) W465-W469.

Dobzhansky T. (1970). Genetics of the evolutionary process. Columbia University Press, New York.

Dowton M, Austin, AD and Antolin MF (1998). Evolutionary relationships among the Braconidae (Hymenoptera : Ichneumonoidea) inferred from partial 16S rDNA gene sequences. Insect Mol. Biol., 7: 129-150.

Dowton M, Belshaw R, Austin AD and Quicke DLJ (2002). Simultaneous molecular and morphological analysis of braconid relationships (Insecta: Hymenoptera: Braconidae) indicates independent mt-tRNA gene inversions within a single wasp family. J. Mol. Evol. 54: 210-226.

Drancourt M, Bollet C, Carlioz A, Martelin R, Gayral JP, and Raoult D (2000). 16S Ribosomal DNA Sequence Analysis of a Large Collection of Environmental and Clinical Unidentifiable Bacterial Isolates J. Clin. Micro. 38:3623–3630.

Dutheil J, Gaillard S, Bazin E, Gléminn S, Ranwez V, Galtier N and Belkhir K (2006). Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. BMC Bioinf. 7:188.

Eddy S (1998). Profile hidden Markov models. Bioinf Rev 14:755-763.

Edgar RC (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucl. Acids Res. 32: 1792-1797.

Farris JS (1989). The retention index and the rescaled consistency index. Cladistics 5:417-419.

Farris JS (1999). Likelihood and inconsistency. Cladistics 15:199-204.

Felsenstein J (1978). Cases in which parsimony or compatibility methods will be positivly misleading. Syst Zool 27:401-410.

Felsenstein J (1985). Confidence limits on phylogenies: An approach using the bootstrap. Evolution 39:783–791.

Felsenstein J (1989) PHYLIP – Phylogeny inference package (version 3.2). Cladistics 5:164-166.

Fitch WM (1970). Distinguishing homologous from analogous proteins. Syst. Zool. 19:99-113.

Fitch WM (1971a). Toward defining the course of evolution: minimum change for a specific tree topology. Syst. Zool. 20:406-416.

Fitch WM (1971b). The nonidentity of invariable positions in the cytochromes c of different species. Biochem. Genet. 5:231–241.

Fitch WM (1971c). Rate of change of concomitantly variable codons, J. Mol. Evol. 1:84-96.

Galtier N (2001). Maximum-likelihood phylogenetic analysis under a covarion-like Model. Mol. Biol. Evol. 18:866-873.

Gaucher E and Miyamoto M (2005). A call for likelihood phylogenetics even when the process of sequence evolution is heterogenous. Mol. Phylog. Evol. 37:928-931.

Gaunt M and Miles M (2002). An insect molecular clock dates the origin of the insects and accords with palaeontological and biogeographic landmarks. Mol. Biol. Evol. 19:748-761.

Gatesy J, Matthee C, Desalle R and Hayashi C (2002). Resolution of a supertree / Supermatrix Paradox. Syst. Biol. 51(4):652–664.

Gatesy J, Baker RH and Hayashi C (2004). Inconsistencies in arguments for the supertree approach: supermatrices versus supertrees of Crocodylia. Syst Biol. 53:342-55.

Gillespie JJ, Yoder MJ, and Wharton RA (2005). Predicted secondary structure for 28S and 18S rRNA from Ichneumonoidea (Insecta: Hymenoptera: Apocrita): Impact on sequence alignment and phylogeny estimation. J. Mol. Evol. 6, 114-137.

Goldman N (1993). Statistical tests of models of DNA substitution. J. Mol. Evol. 36: 182–198.

Goldman N, Anderson JP and Rodrigo AG (2000) Likelihoodbased tests of topologies in phylogenetics. Syst. Biol. 49, 652–670.

Goloboff PA (2002). Techniques for analyzing large data sets. In: DeSalle R., Giribet G.,

Wheeler W., editors. Techniques in molecular systematics and evolution. Basel: Brikhäuser Verlag; p. 70-79.

Goloboff P, Farris J and Nixon K (2003) T.N.T. Tree Analysis Using New Technology. Program and documentation, available at http://www.zmuc.dk/public/phylogeny/tnt.

Goloboff P, Farris J and Nixon K (2008). TNT, a free program for phylogenetic analysis. Cladistics 24(5):774–786.

Goloboff P, Catalano SA, Mirande JM, Szumik CA, Arias JS, Källersjö M and Farris J (2009). Phylogenetic analysis of 73 060 taxa corroborates major eukaryotic groups Cladistics 25:211–230.

Graybeal A (1998). Is it better to add taxa or characters to a difficult phylogenetic problem? Syst Biol 47:9-17.

Grebennikov VV and Scholtz CH (2004). The basal phylogeny of Scarabaeoidea (Insecta:Coleoptera) inferred from larval morphology. Inv. Syst. 18: 321-348.

Gribaldo S, Casane D, Lopez P and Philippe H (2003). Functional divergence prediction from evolutionary analysis: A case study of vertebrate haemoglobin. Mol. Biol. Evol. 20:1754-1759.

Hajibabaei M, Singer G and Hickey D (2006) Benchmarking DNA barcodes. Genome, 49, 851–854.

Hajibabaei M, Singer GAC, Hebert PDN, Hickey DA (2007). DNA barcoding: how it complements taxonomy, molecular phylogenetics and population genetics. Trends Genet 23: doi:10.1016/j.tig.2007.02.001.

Hasegawa M and Hashimoto T (1993). Ribosomal RNA trees misleading? Nature 361:23.

Heath T, Hedtke S and Hillis D (2008). Taxon sampling and the accuracy of phylogenetic analyses. J. Syst. Evol. 46(3): 239-257.

Hebert PDN, Cywinska A, Ball SL and deWaard JR (2003). Biological identications through DNA barcodes. Proc. R. Soc. Lond. B 270,313–321.

Hebert PDN, Ratnasingham, S and deWaard JR (2003). Barcoding animal life: cytochrome c oxidase subunit 1 divergences among closely related species. Proc. Roy. Soc. Lond. B. Biol. (Suppl. 27), S96–S99.

Hebert PDN, Stoeckle MY, Zemlak TS, Francis CM. (2004). Identification of Birds through DNA Barcodes. PLOS Biol. 2:e312.

Hebert PDN and Gregory TR (2005). The Promise of DNA Barcoding for Taxonomy. Syst. Biol. 54:852–859.

Highton R and Peabody RB. (2000). Geographical protein variation and speciation in salamanders of the Plethodon jordani and Plethodon glutinosus complexes in the southern Appalachian mountains with the description of four new species. Pp. 31–94 in R. C. Bruce, R. G. Jaeger, and L. D. Houck, eds. The biology of plethodontid salamanders Kluwer Academic/Plenum Publishers, New York.

Hillis D (1987). Molecular versus morphological approaches to systematics. Ann. Rev. Ecol. Syst. 18:23-42.

Hillis D (1996). Inferring complex phylogenies. Nature 383:130-131.

Hillis D (1998). Taxonomic sampling, phylogenetic accuracy, and investigator bias. Syst. Biol. 47:3-8.

Ho SYW, Phillips MJ, Cooper A and Drummond, AJ (2005). Time dependency of molecular rate estimates and systematic overestimation of recent divergence times. Mol. Biol. Evol. 22:1561-1568.

Hope A (1968) A simplified Monte Carlo significance test procedure. J. Roy. Statist. Soc. B 30:582-598.

Howland D and Hewitt G (1995). Phylogeny of the Coleoptera based on mitochondrial Cytochrome Oxidase I sequence data. Insect Mol. Biol. 4:203-215.

Hunt T (2007). PhD Thesis. Imperial College London, Division of Biology.

Hunt T, Bergsten J, Levkanicova Z, Papadopoulou A, St. John O, Wild R, Hammond PM, Ahrens D, Balke M, Caterino MS, Gómez-Zurita J, Ribera I, Barraclough T, Bocakova M, Bocak L, Vogler A. (2007). A Comprehensive Phylogeny of Beetles Reveals the Evolutionary Origins of a Superradiation.Science 318:1913-1916.

Hunt T and Vogler A (2008). A protocol for large-scale rRNA sequence analysis: Towards a detailed phylogeny of Coleoptera. Mol. Phylog. Evol. 47:289-301.

Jablonski D and Finarelli JA (2009). Congruence of morphologically-defined genera with

molecular phylogenies. PNAS 106:8262-8266.

Jones M and Blaxter M (2006). TaxMan: a taxonomic database manager. BMC Bioinformatics , 7:536.

Jones OR, Purvis A, Baumgart E and Quicke DLJ (2009). Using taxonomic revision data to estimate the geographic and taxonomic distribution of undescribed species richness in the Braconidae (Hymenoptera: Ichneumonoidea). Insect Cons. & Div. 2, 204-212.

Huelsenbeck JP and Bull JJ (1996). A Likelihood Ratio Test to Detect conflicting Phylogenetic Signal. Syst. Biol. (1996) 45 (1): 92-98.

Kallersjo M, Albert VA, Farris JS (1999). Homoplasy increases phylogenetic structure. Cladistics. 15:91–93.

Katoh K, Misawa K, Kuma K and Miyata T (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucl. Acids Res. 30: 3059-3066.

Kauff F, Cox C, and Lutzoni F (2007). WASABI: An Automated Sequence Processing System for Multigene Phylogenies. Syst. Biol. 56:523-531.

Katoh K and Toh H (2008). Recent developments in the MAFFT multiple sequence alignment program. Brief Bioinform 9: 286-298.

Kishino H and Hasegawa M (1989). Evaluation of the Maximum Likelihood Estimate of the Evolutionary Tree Topologies from DNA Sequence Data, and the Branching Order in Hominoidea. J. Mol. Evol. 29:170-179.

Kitching IJ, Forey PL, Humphries CJ and Williams DM (1998). Cladistics: The Theory and Practice of Parsimony Analysis 2nd Edition. Oxford: Oxford University Press. p228.

Kolaczkowski B and Thornton J (2004). Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous. Nature 431:980-984.

Klopfstein S, Kropf C. and Quicke, DLJ (2010). An evaluation of phylogenetic informativeness profiles and the molecular phylogeny of Diplazontinae (Hymenoptera, Ichneumonidae). Syst. Biol. (in press).

Knowles LL and Carstens BC (2007). Delimiting species without monophyletic gene trees. Syst. Biol. 56:887–895.

Knudsen B and Miyamoto MM (2001). A likelihood ratio test for evolutionary rate shifts and functional divergence among proteins. PNAS 98:14512-14517.

Lartillot N, Brinkmann H and Philippe H (2007). Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. BMC Evol. Biol. 7(Suppl 1): S4.

Laurenne NM, Broad GR and Quicke DLJ (2006). Direct optimization and multiple alignment of 28S D2-3 rDNA sequences: problems with indels on the way to a molecular phylogeny of the cryptine ichneumon wasps (Insecta: Hymenoptera). Cladistics, 22, 442-473.

Lee C, Grasso C and Sharlow MF (2002). Multiple sequence alignment using partial order graphs. Bioinf. 18:452-464.

Lester A (2005).WWW::Mechanize - handy web browsing in a perl object. http://search.cpan.org/ petdance/WWW-Mechanize-1.16/lib/WWW/Mechanize.pm,.

Lopez P, Casane D, Philippe H. (2002). Heterotachy, an important process of protein evolution. Mol Biol Evol 19:1–7.

Lunt D, Zhang D-X Szymura JM and Hewitt GM (1996). The insect Cytochrome Oxidase I gene: evolutionary patterns and conserve primers for phylogenetic studies. Insect Mol. Biol. 5:153-165.

Martinez-Navarro E, Galian J and Serrano J (2005) Phylogeny and molecular evolution of the tribe Harpalini (Coleoptera, Carabidae) inferred from mitochondrial cytochrome-oxidase I. Mol. Phylog. Evol. 35:127-146.

Marshall JC, Arevalo E, Benavides E, Sites J and Sites, Jr. JW (2006). Delimiting species: comparing methods for mendelian characters using lizards of the *Sceloporus grammicus* (squamata: phrynosomatidae) complex. Evolution, 60:1050–1065.

McMahon MM and Sanderson MJ (2006). Phylogenetic supermatrix analysis of genbank sequences from 2228 Papilionoide legumes. Syst Biol 55:818-836.

Malia Jr. MJ, Lipscomb DL and Allard MW (2003). The misleading effects of composite taxa in supermatrices. Mol. Phylog. Evol. 27:522-527.

Mayden RL (1997). A hierarchy of species concepts: The denouement in the saga of the

species problem. Pages 381–424 in Species: The units of biodiversity (M. F. Claridge, H. A. Dawah, and M. R.Wilson, eds.).Chapman and Hall, London.

Meier R, Shiyang K, Vaidya G and Ng P (2006). DNA Barcoding and Taxonomy in Diptera: A Tale of High Intraspecific Variability and Low Identification Success. Syst. Biol. 55:715-728.

Misof B, Rickert AM, Buckley T, Fleck G, and Sauer K (2001) Phylogenetic Signal and its Decay in Mitochondrial SSU and LSU rRNA Gene Fragments of Anisoptera. Mol. Biol. Evol. 18:27–37.

Murphy WJ, Eizirik E, Johnson WE, Zhang Y, Ryder O and O'Brien S (2001). Molecular phylogenetics and the origins of placental mammals. Nature 409:614–618.

Nielsen R and Matz M (2006). Statistical Approaches for DNA Barcoding. Syst. Biol. 55:162-169

Nilsson RH, Ryberg M, Kristiansson E, Abarenkov K, Larsson KH and Kõljalg U (2006). Taxonomic reliability of DNA sequences in public sequence databases: a fungal perspective. PLoS ONE 1:e59.

Nixon KC, and Wheeler Q (1990). An amplification of the phylogenetic species concept. Cladistics 6:211–223.

Notredame C (2007). Recent Evolutions of Multiple Sequence Alignment Algorithms PLoS Comput Biol. August; 3:e123.

O'Meara BC. (2010) New heuristic methods for joint species delimitation and species tree inference. Syst. Biol. 2010 59:59-73.

Ousterhout JK (1998). Scripting: Higher-level programming for the 21st century Computer. 31: 23-30.

Paradis E, Claude J and Strimmer K (2004). APE: Analyses of Phylogenetics and Evolution in R language. Bioinf. 20:289-290.

Philippe H and Lopez P (2001). On the conservation of protein sequences in evolution. Trends Biochem. Sci. 26:414–416.

Phillips MJ, Delsuc F and Penny D (2004). Genome-scale phylogeny and the detection of systematic biases. Mol. Biol. Evol. 21:1455-1458.

Pocock M, Down T and Hubbard T (2000). BioJava: Open Source Components for Bioinformatics. ACM SIGBIO Newsletter 20(2):10-12.

Poe S (1998). Sensitivity of phylogeny estimation to taxonomic sampling. Syst. Biol. 47:18-31.

Poe S (2003). Evaluation of the strategy of long-branch subdivision to improve the accuracy of phylogenetic methods. Syst. Biol. 52:423–428.

Pollock DD and Bruno WJ (2000). Assessing an Unknown Evolutionary Process: Effect of Increasing Site-Specific Knowledge Through Taxon Addition. Mol. Biol. Evol. 17:1854–1858.

Pons J, Barraclough T, Gomez-Zurita J, Cardoso A, Duran DP, Hazell S, Kamoun S, Sumlin W and Vogler A (2006). Sequence-Based Species Delimitation for the DNA Taxonomy of Undescribed Insects. Syst. Biol. 55:595–609.

Pupko T, Huchon D, Cao Y, Okada N and Hasegawa M (2002). Combining multiple data sets in a likelihood analysis: Which models are the best? Mol. Biol. Evol. 19:2294-2307.

Purvis A and Quicke DLJ (1997). Phylogeny reconstruction: Are the big easy? TREE 12:49-50.

Qiu Y, Lee J, Bernasconi-Quadroni F, Soltis DE, Soltis PS, Zanis M, Zimmer EA, Chen Z, Savolainen V and Chase M. (1999). The earliest angiosperms: evidence from mitochondrial, plastid and nuclear genomes. Nature 402:404–407.

Quicke DLJ and van Achterberg C (1990). Phylogeny of the subfamilies of the family Braconidae (Hymenoptera, Ichneumonoidea). Zoologische Verhandelingen, 258: 1-95.

Quicke DLJ and Belshaw R (1999). Incongruence Between Morphological Data Sets: An Example from the Evolution of Endoparasitism Among Parasitic Wasps (Hymenoptera: Braconidae). Syst. Biol. 48:436-454.

Quicke DLJ, Laurenne NM, Fitton MG and Broad GR (2009). A thousand and one wasps: a 28S rDNA and morphological phylogeny of the Ichneumonidae (Insecta: Hymenoptera) with an investigation into alignment parameter space and elision. J. Nat. Hist. 43:1305-1421.

R Development Core Team (2007). R: A language and environment for statistical computing. R Foundation for Statistical Computing.

Rambaut A and Grassly N (1997). Seq-gen: an application for the Monte Carlo

simulation of DNA sequence evolution along phylogenetic trees. Bioinf. 13:235-238.

Rannala B, Huelsenbeck J, Yang Z and Nielsen R (1998). Taxon Sampling and the Accuracy of Large Phylogenies. Syst. Biol. 47: 702–710.

Roe A and Sperling F (2007). Patterns of evolution of mitochondrial cytochrome c oxidase I and II DNA and implications for DNA barcoding. Mol. Phylog. Evol. 44:325-345.

Roff D and Bentzen P (1989). The statistical analysis of mitochondrial DNA polymorphisms: $\chi 2$ and the problem of small samples. Mol. Biol. Evol. 6:539-545.

Rokas A, Nylander J, Ronquist F and Stone G (2002). A Maximum-Likelihood Analysis of Eight Phylogenetic Markers in Gallwasps (Hymenoptera: Cynipidae): Implications for Insect Phylogenetic Studies. Mol. Phylog. Evol. 22:206–219.

Ronquist F and Huelsenbeck J (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinf 19:1572–1574.

Royston P (1982). Algorithm AS 181: The W test for normality. Appl. Stat. 31:176-180.

Ruano-Rubio V and Fares M (2007). Artifactual phylogenies caused by correlated distribution of substitution rates among sites and lineages: the good, the bad and the ugly. Syst. Biol. 56:68-82.

Ryberg M, Nilsson RH, Kristiansson E, Töpel M, Jacobsson S, Larsson E. 2008. Mining metadata from unidentified ITS sequences in GenBank: a case study in Inocybe (Basidiomycota). BMC Evol. Biol. 8: 50–64.

Rzhetsky A and Nei M (1994). Unbiased estimates of the number of nucleotide substitutions when substitution rate varies among different sites. J. Mol. Evol. 38:295–299.

Sanderson MJ, Driskell AC, Ree RH, Eulenstein O and Langley S (2003). Obtaining maximal concaternated phylogenetic data sets from large sequence databases. Mol. Biol. Evol. 20:1036-1042.

Sanderson MJ, Boss D, Chen D, Cranston K and Wehe A (2008). The PhyLoTA Browser: Processing GenBank for Molecular Phylogenetics Research. Syst. Biol. 57:335–346.

Saraste M (1990). Structural features of cytochrome oxidase. Q. Rev. Biophys. 23:331-366.

Savolainen V, Cowan RS, Vogler AP, Roderick GK and Lane R (2005). Towards writing the encyclopedia of life, an introduction to DNA barcoding. Proc. Royal Soc. B, 360:1805–1811.

Sharp PM, Shields DC, Wolfe KH and Li W (1989). Chromosomal location and evolutionary rate variation in enterobacterial genes. Science 258: 808–810.

Sheldon FH and Bledsoe AH (1993). Avian moolecular systematics, 1970's to 1990's. Annu. Rev. Ecol. Syst. 24:243-278.

Smith MA, Fernandez-Triana J, Roughley R and Hebert P (2009). DNA barcode accumulation curves for understudied taxa and areas. Mol. Ecol. Resources 9 (Suppl. 1), 208–216.

Shimodaira H (2002). An Approximately Unbiased Test of Phylogenetic Tree Selection. Syst. Biol. (2002) 51:492-508.

Spicer GS (1995). Phylogenetic utility of the mitochondrial cytochrome oxidase gene: molecular evolution of the Drosophila buzzatii species complex. J. Mol. Evol. 41:749-759.

Stajich J, Block D, Boulez K, Brenner S, Chervitz S, Dagdigian C, Fuellen G, Gilbert J, Korf I, Lapp H, Lehväslaiho H, Matsalla C, Mungall C, Osborne B, Pocock M, Schattner P, Senger M, Stein L, Stupka E, Wilkinson M and Birney E (2002). The Bioperl toolkit: Perl modules for the life sciences. Genome Res. 12:1611-1618.

Stamatakis A, Ludwig T and Meier H (2005). RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinf. 21:456-463.

Steel M (2005). Should phylogenetic models be trying to ‘fit an elephant’? TIG 21:307-309.

Storey J (2002). A direct approach to false discovery rates. J. Roy. Stat. Soc. B 64:479-498

Sullivan J, Swofford D and Naylor G (1999). The effect of taxon sampling on estimating rate heterogeneity parameters of maximum-likelihood models. Mol. Biol. Evol. 16:1347-1356.

Swofford DL (2000). PAUP*. Phylogenetic Analysis Using Parsimony (*and Other

Methods). Version 4. Sinauer Associates, Sunderland, Massachusetts.

Swofford DL,Waddell PJ, Huelsenbeck JP, Foster P, Lewis PO and Rogers JS (2001). Bias in phylogenetic estimation and its relevance to the choice between parsimony and likelihood methods. Syst. Biol. 50, 525–539.

Tamura K (1992). Estimation of the Number of Nucleotide Substitutions When There Are Strong Transition-Transversion and G+C-Content Biases. Mol. Bid Evol. 9:678-687.

Tateno Y, Takezaki N, and Nei M (1994). Relative Efficiencies of the Maximum-Likelihood, Neighbor-joining, and Maximum-Parsimony Methods When Substitution Rate Varies with Site. Mol. Biol. Evol. 11:26 l-277.

Thompson J, Higgins D and Gibson T (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignments through sequence weighting, position specific gap penalties and weight matrix choice. Nucl. Acids Res. 22:4673-4680.

Thomson R and Shaffer B (2010). Rapid progress on the vertebrate tree of life. BMC Biology, 8:19

Tuffley C and Steel M (1998). Modelling the covarion hypothesis of nucleotide substitution. Math. Biosci. 147:63-91

Van Valen L (1976). Ecological species, multispecies, and oaks. Taxon 25:233–239.

Venditti C, Meade A and Pagel M. (2006). Detecting the node-density artifact in phylogeny reconstruction. Syst. Biol. 55:637-643.

Vogler AP, Cardoso A, Barraclough TG (2005) Exploring rate variation among and within sites in a densely sampled tree: Species level phylogenetics of North American Tiger beetles (Genus Cicindela). Syst Biol 54:4-20.

Wheeler W (1995). Sequence alignment, parameter sensitivity, and the phylogenetic analysis of molecular data. Syst. Biol. 44:321-331.

Whitfield JB (1992). The polyphyletic origin of endoparasitism in the cyclostome lineages of Braconidae (Hymenoptera). Syst. Entom. 17:273-286.

Whitfield JB (2002). Estimating the age of the polydnavirus/braconid wasp symbiosis. PNAS 99:7508-7513.

Whitfield JB, Mardulyn P, Austin AD and Dowton M (2002). Phylogeny of the

Microgastrinae (Hymenoptera: Braconidae) based on multiple genes and morphology. Syst. Entom. 27:337-359.

Wiemers M and Fiedler K (2007). Does the DNA barcoding gap exist? – a case study in blue butterflies (Lepidoptera: Lycaenidae). Front. Zool. 4:8.

Wiens JJ (2003). Missing data, incomplete taxa, and phylogentic accuracy. Syst. Biol. 52:528-538.

Wiens JJ (2007). Species Delimitation: New Approaches for Discovering Diversity Syst. Biol. 56:875–878.

Wiens JJ and Reeder TW (1995). Combining datasets with differetn numbers of taxa for phylogenetic analysis. Syst. Biol. 44:548-558.

Wiens JJ and Penkrot TA (2002). Delimiting Species Using DNA and Morphological Variation and Discordant Species Limits in Spiny Lizards (*Sceloporus*) Syst. Biol. 51:69-91,

Wiley EO (1978). The Evolutionary Species Concept Reconsidered Systematic Zoology 27:17-26.

Wilson JJ (2010) Assessing the Value of DNA Barcodes and Other Priority Gene Regions for Molecular Phylogenetics of Lepidoptera. PLoS ONE 5: e10525. doi:10.1371/journal.pone.0010525

Wong KM, Suchard MA and Huelsenbeck JP (2008). Alignment uncertainty and genomic analysis. Science 5862:473-476.

Wortley AH, Scotland RW (2006). Determining the potential utility of datasets for phylogeny reconstruction. Taxon 55: 431–442.

Will KW and Rubinoff D, (2004). Myth of the molecule: DNA barcodes for species cannot replace morphology for identification and classification. Cladistics 20, 47–55.

Wu J, Susko E, Roger AJ (2008). An independent heterotachy model and its implications for phylogeny and divergence time estimation. Mol. Phylog. Evol. 46:801-806.

Wu D, Hartman A, Ward N and Eisen J (2008). An Automated Phylogenetic Tree-Based Small Subunit rRNA Taxonomy and Alignment Pipeline (STAP). PLoS ONE 3:e2566.

Yang Z (1994). Maximum likelihood phylogenetic estimation from DNA sequences

with variable rates over sites: Approximate methods. J. Mol. Evol. 39:306-314.

Yang Z (1995). A space-time process model for the evolution of DNA sequences. Genetics 139:993–1005.

Yang Z (1996a). Among-site rate variation and its impact on phylogenetic analyses. Trends Ecol. Evol. 11:367–370.

Yang Z (1996b). Maximum-likelihood models for combined analyses of multiple sequence data. J. Mol. Evol. 42:587-596.

Yang Z, Goldman N and Friday A (1994). Comparison of models for nucleotide substitution used in maximum-likelihood phylogenetic estimation. Mol. Biol. Evol. 11: 316–324.

Yang Z, Nielsen R, Goldman N and Pedersen A (2000) Codon-substitution models for heterogeneous selection pressure at amino acid sites. Genetics 155, 431–449.

Zaldivar-Riveróna A, Moria M and Quicke DLJ (2006). Systematics of the cyclostome subfamilies of braconid parasitic wasps (Hymenoptera: Ichneumonoidea): A simultaneous molecular and morphological Bayesian approach. Mol. Phylog. Evol. 38:130-145.

Zardoya R and Meyer A (1996). Phylogenetic performance of mitochondrial protein-coding genes in resolving relationships among vertabrates. Mol. Biol. Evol. 13:933-942.

Zhou Y, Rodrigue N, Lartillot N and Philippe H (2007). Evaluation of the models handling heterotachy in phylogenetic inference. BMC Evol. Biol. 7:206.

Zuckerkandl E and Pauling L. (1965). Evolutionary divergence and convergence in proteins. In Evolving genes and proteins (ed. V. Bryson and H. Vogel), pp. 97–166. Academic Press, New York, NY.

Zwickl D and Hillis D (2002). Increased Taxon Sampling Greatly Reduces Phylogenetic Error. Syst. Biol. 51:588-598.

Appendix 1: Key subroutines. Here I provide code for keys task performed by HIP-DB, and some of the analyses presented. This code runs under the linux environment, and requires installation additional phylogenetic programs. For complete scripts and advice on implementation, contact the author. Note, code should be copied from the electronic version of the thesis, as not to inadvertently introduce inactivating linebreaks.

Appendix 1.1: make_keyfile subroutine, which calls store_nodes, parse_namesfile, traverse_nodes, get_taxstring, and walk_name. Function is to reads the ncbi tax database files and creates taxstrings (see chapter four).

```
sub make_keyfile
{

################################################################################
#
#       parse_ncbi_tax_database.pl
#       perl script by D.Chesters
#       structurally similar to build_taxkeys.pl (James Abbott and Toby Hunt),
#       but simpler, and requires no perl modules or database software.
#
#       first reads nodes.dmp, then names.dmp
#       then traverses nodes from specified start,
#       strings are printed to file during the process, to file such as: key_Jan2010_Coleoptera
#       then taxon counts printed to log
#
################################################################################

#  06NOV2009 given names are by the filtered method (used to generate strings) rather than raw
#  12JAN2010 option to ignore subsp

#$starting_node = 75546;

# Sample taxonomy IDS:
# Eukaryota =    2759     , Insecta = 50557, Endopterygota   = 33392,         Coleoptera = 7041,

$ignore_atypical_ranks_to_shorten_codes = 1;        # 0==no. 1==yes
$ignore_subspecies                        = 1;


system("rm key_Jan2010");


# globals
@all_nodes;
%nodes_taxbuild;
%all_taxstrings;
%code_counts;
%nonbinomial_recorder;
%binomial_recorder;
%unidentified_sp_hash;

$date = localtime time;
$month = $date;
$month=~ s/\w+\s+(\w+)\s+\d+\s+\S+\s+(\d+)$/$1$2/;
$month= "Jan2010";
print "month:$month\n";

open(LOG, ">>locus_contribution_LOG") || die "cant open logfile\n";

print LOG "\nsub parse_ncbi_tax_database.pl\n$date\nstarting_node:$starting_node
ignore_atypical_ranks_to_shorten_codes:$ignore_atypical_ranks_to_shorten_codes\n";


###############
store_nodes();#
###############

###################
parse_namesfile();#
###################
```

```
$starting_name = $nodes_taxbuild{$starting_node}{name};$starting_name_ff = $starting_name;

$starting_name =~ s/^(\w).+$/$1/;

$key_file = "key_" . $month . "_$starting_name_ff";
open(OUT ,">$key_file") || die "cant open \n";
$key_file2 = "key_" . $month;
open(OUT2 ,">$key_file2") || die "cant open \n";



################################################
traverse_nodes($starting_node , $starting_name);#
################################################

close(OUT);close(OUT2);

@keys = keys %how_many;@keys = sort @keys;

print LOG "\nfrom node:$starting_name_ff\n";
print "\nfrom node:$starting_name_ff\n";

foreach $key(@keys)
{
 print LOG "$key\t" , $how_many{$key} , "\n";
 print "$key\t" , $how_many{$key} , "\n";

}



print "\nend of script\n";
print LOG "\nend of script\n";
close(LOG);

my @keys = keys %binomial_recorder;@keys = sort @keys;

$count_binomials = 0;
$count_binomials2 = 0;

foreach $key(@keys)
{
$count_binomials2 ++;

 print "$key\t" , $binomial_recorder{$key} , "\n";
$count_binomials += $binomial_recorder{$key};
}

print "\ncount_binomials:$count_binomials $count_binomials2\n";

}



sub store_nodes
{


open(NODES , "nodes.dmp") || die "cant open nodes\n";

my $line_counter=0;
while (my $line = <NODES>)
        {
        if($line =~ /^(\d+)\t[\|]\t([^\t]+)\t[\|]\t([^\t]+)\t[\|]\t/)
                {
                my $tax_id = $1;my $parent_tax_id = $2;my $rank = $3;
                $rank_hash{$rank}++;
#               print "tax_id:$tax_id parent_tax_id:$parent_tax_id rank:$rank\n";

                my $current_rankcode="";
                 #
                                  if($rank eq "suborder" ){$current_rankcode=1};
                 if($rank eq "infraorder"){$current_rankcode=2};
                 if($rank eq "series" ){$current_rankcode=2};
                 if($rank eq "superfamily" ){$current_rankcode=3};
                 if($rank eq "family"){$current_rankcode=4};
                 if($rank eq "subfamily"){$current_rankcode=5};
                 #if($rank eq "tribe" || $tribe_test=~
/^\w+ini$/){$current_rankcode=6;$current_tribe=$node->scientific_name();$current_tribe=~ s/ /_/g};
                 if($rank eq "tribe"){$current_rankcode=6};
                 # if($rank eq "subtribe" ){$current_rankcode=""};
```

```
                 if($rank eq "genus"){$current_rankcode="7"};
                 # if($rank eq "species"){$current_rankcode="";$in_species=1;}else{$in_species=0; #
$current_species=$node->scientific_name()


                $nodes_taxbuild{$tax_id}{rank} = $rank;
                $nodes_taxbuild{$tax_id}{rank_code} = $current_rankcode;

                $nodes_taxbuild{$tax_id}{parent} = $parent_tax_id;
                $nodes_taxbuild{$parent_tax_id}{child_nodes} .= "\t" . $tax_id;

                }else{
                print "line_counter:$line_counter line:$line";
                die "UNEXPECTED LINE:$line\nquitting\n";
                }
        $line_counter++;
        }

close(NODES);

my @ranks = keys %rank_hash;@ranks = sort @ranks;
print "ranks found in nodes.dmp:\n";
print LOG "ranks found in nodes.dmp:\n";

foreach my $rank(@ranks){print "$rank\t" , $rank_hash{$rank} , "\n";print LOG "$rank\t" ,
$rank_hash{$rank} , "\n"};

@all_nodes = keys %nodes_taxbuild;@all_nodes = sort @all_nodes;


}


sub parse_namesfile
{

# here just parse the scientific name of each node. ignore synonyms etc

open(NAMES , "names.dmp") || die "cant open names\n";

my $line_counter=0;
while (my $line = <NAMES>)
        {

# 24    |       Shewanella putrefaciens   |               |       scientific name  |

        if($line =~ /^(\d+)\t[\|]\t([^\t]+)\t[\|]\t([^\t]*)\t[\|]\tscientific name/)
                {
                my $tax_id = $1;my $name = $2;#my $rank = $3;
                # print "tax_id:$tax_id name:$name\n";
                $nodes_taxbuild{$tax_id}{name} = $name;
                }else{
                if($line =~ /^(\d+).+scientific name/){die "UNEXPECTED LINE:\n$line\nquitting\n"}
                }

        }



}



sub traverse_nodes
{
my $current_node = $_[0];my $current_node_taxstring = $_[1];

# my $current_node = shift;

my $child_nodes = $nodes_taxbuild{$current_node}{child_nodes};$child_nodes =~ s/^\t//;
my @child_nodes_array = split(/\t/, $child_nodes);
 print "\ncurrent_node:$current_node child_nodes:$child_nodes\n";

foreach my $child(@child_nodes_array)
        {
#       my $name = $nodes_taxbuild{$child}{name};
        my $rank = $nodes_taxbuild{$child}{rank};$rank =~ s/\s/_/g;$how_many{$rank}++;
        # my $current_rankcode = $nodes_taxbuild{$child}{rank_code};

        ##############################################################################
       my $taxstring_and_name = get_taxstring( $child , $current_node_taxstring);#
```

```
        ###############################################################################


        my @taxstring_array = split ( /__/, $taxstring_and_name );
        my $taxstring = $taxstring_array[0];
        my $name = $taxstring_array[1];

#       print "taxstring:$taxstring name:$name\n";

        my $name_copy = $name;$name_copy =~ s/\s/_/g;

        my $child_complete_lineage = $nodes_taxbuild{$current_node}{complete_lineage} . "$rank:$name
";
        my $child_taxstring = $current_node_taxstring . $taxstring;

        $nodes_taxbuild{$child}{complete_lineage} = $child_complete_lineage;


        if(length($child_taxstring)>=30){
        print "NAME TOO LONG:\nnode:$current_node child:$child name:$name
child_taxstring:$child_taxstring lin:" , $nodes_taxbuild{$child}{complete_lineage} , "\n";
        print LOG "NAME >=30chars:\nnode:$current_node child:$child name:$name
child_taxstring:$child_taxstring lin:" , $nodes_taxbuild{$child}{complete_lineage} , "\n";

                }


#CA1D3D4C5Co6Co7pay Colymbetes_paykulli 183353 species suborder:Adephaga superfamily:Dytiscoidea
family:Dytiscidae subfamily:Colymbetinae tribe:Colymbetini genus:Colymbetes
species:Colymbetes_paykulli
        print OUT "$child_taxstring $name_copy $child $rank $child_complete_lineage\n";
        print OUT2 "$child_taxstring $name_copy $child $rank $child_complete_lineage\n";
        # print "$child_taxstring $name_copy $child $rank $child_complete_lineage\n";



        ##########################################
        traverse_nodes($child , $child_taxstring);#
        ##########################################

        }


}

sub get_taxstring
{
my $current_node = $_[0];my $current_node_taxstring = $_[1];#my $current_rankcode= $_[2];
my $name = $nodes_taxbuild{$current_node}{name};
my $rank = $nodes_taxbuild{$current_node}{rank};
my $rank_code = $nodes_taxbuild{$current_node}{rank_code};
my $newcode;
my $to_walk;

my $truncate_factor = 1;
if($rank eq "genus"){$truncate_factor =2};
if($rank eq "species" || $rank eq "subspecies"){$truncate_factor =3};




# get rid of wierd chars

$name =~ s/\-//g;
$name =~ s/\'(\w+)\'/$1/g;
$name =~ s/\.//g;

my @words = split( / /, $name );

   if ( $rank eq "species" )
        {

if ( $name =~ /^[A-Z][a-z]+\s+[a-z]+$/){$binomial_recorder{$name}++}else{$nonbinomial_recorder{$name}
++}

        $name=~ s/BMNH//;
        my @newwords;
        if ( $#words >= 2 ) #  THREE or more words
                {

                foreach my $word (@words)
```

```
                        {
                unless($word =~ /^sp$|^aff$|^nr$|^cf$|^spp$/ ){push( @newwords, $word )};
                        }

                $to_walk = $newwords[$#newwords];
                if($to_walk =~ /\d/ && length($to_walk)<=10){$truncate_factor = length($to_walk)};
                $newcode = walk_name(     $to_walk , $current_node_taxstring , $truncate_factor ,
$current_node);

        }else
                {

               if ( $#words == 1 ) #  TWO words
                        {

                        $to_walk = $words[1];
                $newcode = walk_name(     $to_walk , $current_node_taxstring , $truncate_factor ,
$current_node);
                        #if($to_walk =~ /^sp$|^aff$|^nr$|^cf$|^spp$/ ){$unidentified_sp_hash{
$words[0] . "__" . $words[1]}++ }

                        }else{

                        $to_walk = $words[0];
                        $newcode = walk_name(     $to_walk , $current_node_taxstring ,
$truncate_factor , $current_node);

                        }

        }

        }
        elsif ( $rank eq "subspecies" )
        {


                $to_walk = $words[$#words];
                $newcode = walk_name(     $to_walk , $current_node_taxstring , $truncate_factor ,
$current_node , 1);

        }else{   #if not species or subspecies

        # if you want to shorten codes, could tell it to ignore atypical ranks here, such as subgenus,
species complex, no rank, etc

#         print "passing n:$name  c:$current_node_taxstring to subwalkname\n";


        if($ignore_atypical_ranks_to_shorten_codes==0)
                {
                $to_walk = $name;
               $newcode = walk_name($to_walk , $current_node_taxstring , $truncate_factor ,
$current_node);
                }else{

                if($rank eq "subspecies" || $rank eq "species" || $rank eq "genus" || $rank eq
"tribe" || $rank eq "subfamily" || $rank eq "family" || $rank eq "superfamily" || $rank eq "suborder"
|| $rank eq "order" )
                        {
                        $to_walk = $name;
                $newcode = walk_name($to_walk , $current_node_taxstring , $truncate_factor ,
$current_node);
                        }

                }
        }

        my $newcode_plus_name = $newcode . "__" . $to_walk;
        return ($newcode_plus_name);

}




sub walk_name
{
my $to_walk = $_[0];my $current_node_taxstring = $_[1];my $truncate_factor= $_[2];my $current_node =
$_[3];my $subsp_test = $_[4];
my $name = $nodes_taxbuild{$current_node}{name};
my $current_rankcode = $nodes_taxbuild{$current_node}{rank_code};
```

```
# print "current_node:$current_node current_rankcode:$current_rankcode\n";

my $current_index = 0;
my $test_this_string;

        if($ignore_subspecies==1 && $subsp_test == 1)
                {
                $test_this_string = $current_node_taxstring;
                $test_this_string =~ s/$current_node_taxstring(.*)$/$1/;

                }else{

$test_this_string = $current_node_taxstring . substr($to_walk , 0,
($truncate_factor+$current_index));

while(exists($all_taxstrings{$test_this_string}))
        {
        $current_index++;
        if($current_index<=100)
                {
                $test_this_string = $current_node_taxstring . substr($to_walk , 0,
($truncate_factor+$current_index));
                }else{
                if($test_this_string =~ /^(.+[_])(\d+)$/){$test_this_string = $1 .
($2+1)}else{$test_this_string = $test_this_string . "_1" }
                }
        };

$all_taxstrings{$test_this_string} = 1;
# my $to_return = substr($to_walk , 0,  ($truncate_factor+$current_index));
# my $to_return = $test_this_string; NO! this is the total string, we need to return just the new
substring
$test_this_string =~ s/$current_node_taxstring(.+)$/$1/;

                }
## hash out this line if you dont want rank numbers in tax strings:
$test_this_string .= $current_rankcode;
return($test_this_string);
}
```

Addendix 1.2: do_blast_searches. Uses BioPerl code to blast a database and extract hits. See chapter three.

```
sub do_blast_searches {

$database_name = "database_fasta_Nov2009.parsed";
$which_database = $database_name;

my %start_hash = (); # these must be reset for each search
my %end_hash = ();
my %strand_hash = ();
my $query_length = "";
my $current_query_file = "";
my %cutoff_name_hash = ();

        if($rapid_blast==1)
                {$current_query_file =  "~/hip-db/blast_searches/" . $current_gene .
"REDUCED.aligned_edited";
                }else{$current_query_file =  "~/hip-db/blast_searches/" . $current_gene .
".aligned_edited"};

                my $current_blast_outfile = "" .$current_gene . ".blast.out"; # ~/hip-
db/blast_searches/

        system("rm $current_blast_outfile");

                # run local blast. should be in the path
        my $command = "blastall -p blastn -d $which_database -i $current_query_file -o
$current_blast_outfile $blast_search_evalue -v 900000 -b 900000";
        print "entering shell command:\n$command\n";print LOG "entering shell command:\n$command\n";
        system($command);

        print "$current_gene blast search completed. parsing results file\n";print LOG "$current_gene
blast search completed. parsing results file\n";
        my $in = new Bio::SearchIO        (       -format => "blast",
                                                  -file =>         "$current_gene.blast.out" #
blast_searches/
                                          );

                # use BioPerl parser for the blast output
        my $count_hits2=0;print "\nhit_no current_id start end strand\n";
        my $number_queries=0;my $query_length_sum = 0;

        while (my $result = $in->next_result)
                {
                $query_length = $result->query_length;$query_length_sum +=
$query_length;$number_queries++;
                while (my $hit = $result->next_hit)
                        {
                        while (my $hsp = $hit->next_hsp)
                                {

                                my $current_id = $hit->name;my $current_strand = $hsp->strand(hit);
        # print " length:" , $hsp->length(hit);    print " eval:" , $hsp->expect;
                                my $total_length_of_hit = $hit->length;
                                if($total_length_of_hit >= $trim_cutoff)
                                        {
                                        # print "current_id:$current_id
total_length_of_hit:$total_length_of_hit over $trim_cutoff\n";
                                        $cutoff_name_hash{$current_id} = 1;
                                        };

                                my $current_start = $hsp->start(hit);my $current_end = $hsp-
>end(hit);
                                $count_hits2++;if($count_hits2=~ /000$/){print "$count_hits2
$current_id $current_start $current_end $current_strand\n"};

                                if($current_strand == 1)
                                        {

                                # some loci are submitted in opposite strand, so record strand as
well as start and end position of hit.
                                # the start and end position of the hit to a given id are recorded
seperatly.
                                # as multiple blast searches are being performed, the lowest start
position (of all hits to the id)
                                # and highest end position is taken.
```

```
                                        if(length($start_hash{$current_id}) >= 1)
                                                {if($start_hash{$current_id} >=
$current_start){$start_hash{$current_id} = $current_start};
                                                }else{$start_hash{$current_id} = $current_start};

                                        if(length($end_hash{$current_id}) >= 1)
                                                {if($end_hash{$current_id} <=
$current_end){$end_hash{$current_id} = $current_end};
                                                }else{$end_hash{$current_id} = $current_end};
                                        $strand_hash{$current_id} = 1;
                                        }else{

                                        if(length($start_hash{$current_id}) >= 1)
                                                {if($start_hash{$current_id} >=
$current_start){$start_hash{$current_id} = $current_start};
                                                }else{$start_hash{$current_id} = $current_start};

                                        if(length($end_hash{$current_id}) >= 1)
                                                {if($end_hash{$current_id} <=
$current_end){$end_hash{$current_id} = $current_end};
                                                }else{$end_hash{$current_id} = $current_end};

                                        $strand_hash{$current_id} = 2;
                                        };

                                };
                        };
                };
$query_length = $query_length_sum / $number_queries;$query_length = int($query_length);
print "mean_query_length:$query_length\n";
my $hit_limit = $query_length * 0.6;
print "results file parsed, $count_hits2 hits (total, incl repeats). \nfetching sequences with hits >
$hit_limit (query_length:$query_length)\n";
print LOG "results file parsed, $count_hits2 hits (total, incl repeats).
\nmean_query_length:$query_length number_queries:$number_queries fetching sequences with hits >
$hit_limit\n";

# system("rm $current_blast_outfile");

open(OUT1, ">$current_gene.blast.out.retrieved") || die "cant open
outfile:$current_gene.blast.out.retrieved\n";

                                # now go through all the recorded hits and print if > hit_limit
(currently set for query length * 0.8)
my $count_hits=0; my $count_hits3=0; # start / end position and strand are taken from hash for each
id, then ncbi_fastacmd takes the sequence from
my @all_ids = keys %start_hash;    # a local blast database (use -o T option when making this)
foreach my $current_id(@all_ids)# fastacmd prints extra bits (>lcl|sequence_id No definition line
found), 2 lines below remove these
        {
        my $current_start = $start_hash{$current_id};
        my $current_end = $end_hash{$current_id};
        my $current_length = $current_end-$current_start;
        if ($current_length >= $hit_limit)
                {
                $count_hits++;
                if($count_hits=~ /00$/){print "$count_hits $current_id $current_start
$current_end\n"};


                my $current_strand = $strand_hash{$current_id};
                my $for_log = $current_id . "_" . $current_start . "_" . $current_end . "_" .
$current_strand; print LOG "$for_log ";#print "$for_log\n";
                my $entry_retrieved = "";
#               if($trim_entries == 1)

                if(exists($cutoff_name_hash{$current_id}))
                        {
                        print "trimming $current_id\n";

                        # HACK: will not work under some circumstances:

                        my $addthismanybasestohomologousbit = 100;
                        if($current_start < $current_end){$current_start = $current_start -
$addthismanybasestohomologousbit;$current_end = $current_end + $addthismanybasestohomologousbit}else{
                        $current_start = $current_start +
$addthismanybasestohomologousbit;$current_end = $current_end - $addthismanybasestohomologousbit}

                        $entry_retrieved =`fastacmd -d $which_database -s $current_id -S
$current_strand -L $current_start,$current_end`;
                        }else{
```

```
                        $entry_retrieved =`fastacmd -d $which_database -s $current_id -S
$current_strand`;
                        }


                # $entry_retrieved =~ s/\012\015?|\015\012?/\n/g; just an idea, not tried
                $entry_retrieved =~ s/>lcl\|/>/;
                $entry_retrieved =~ s/\:\d+.\d+\s+No definition line found//;
                $entry_retrieved =~ s/No definition line found//;
                # $entry_retrieved =~ s/(>\S+)\s+(\n)/$1$2/; # this was to prevent the odd spaces
appearing at the end of the id line.
                                                                # but seems to be removing the newline
also, so remove the spaces during parsing of blast.out.retrieved instead
                print OUT1 $entry_retrieved;
        #       print $entry_retrieved;


                }else{$count_hits3++;};
        };

print "found $count_hits hits longer than hit_limit($hit_limit). $count_hits3 too
short\n\n*****************************\n\n";
print LOG "\nfound $count_hits $current_gene hits longer than hit_limit($hit_limit). $count_hits3 too
short\n";

close(OUT1);


        }; # end of sub (do blast searches)
```

Appendix 1.3: sub record_tree_structure_and_prune. Calls generate_newick_string. Pruning algorithms as used to make trees comparable (chapters 2 and 3).

```
sub record_tree_structure_and_prune
        {
my $subinput = shift;

#$pruned_newick_string = record_tree_structure_and_prune($all_loci_tree  . "___" .
$only_other_loci_tree);#

my %list_of_taxa_which_wont_be_pruned_from_added_trees2;
my @subinput_array = split(/___/, $subinput);
my $tree = $subinput_array[0];my $glftt_tree = $subinput_array[1];


#print "\n\ntree:$tree\n\nglftt_tree:$glftt_tree\n\n";
#print "glftt_tree:$glftt_tree\n\n";
#die;

$glftt_tree =~ s/\s//g;$glftt_tree =~ s/:\d+\.\d+(\D)/$1/g;$glftt_tree =~ s/:\d+(\D)/$1/g;

while($glftt_tree =~ /[\(\)\,][^\(\)\,]+[\(\)\,]/)
        {
        $glftt_tree =~ s/([\(\)\,])([^\(\)\,]+)([\(\)\,])/$1$3/;
        my $tha_taxon = $2;
#       print "not gonna prune $tha_taxon ";
        $list_of_taxa_which_wont_be_pruned_from_added_trees2{$tha_taxon} =1;
        }


$tree =~ s/\s//g;
my $tree_copy = $tree;


%nodes = ();

my @test = keys %nodes;

if (scalar @test >= 2){die "error\n@test\n"}
print "sub record_tree_structure\n";

if($tree =~ /[^\)\(\,]:[\d\.]+/){$branchlength=1}else{$branchlength=0};


# assign node numbers to tips,:


my $counter=0;my $x=1;
while ($x == 1)
        {
        $counter++;
        if($branchlength==1)
                {
                $tree_copy =~ s/([\,\(])([^\)\,\(]+):([^\,\)\(]+)/$1$counter/; $current_tip =
$2;$current_length = $3;
                }else{
                $tree_copy =~ s/([\,\(])([^\(\)\,]+)([\,\)])/$1$3/; $current_tip = $2;
                }


        $nodes{$current_tip}{tip_number}  = $counter;
        $nodes{$current_tip}{sum_bl}              = $current_length;
        $nodes{$current_tip}{is_node_a_tip}       = 1;

#        print "counter:$counter current_tip:$current_tip current_length:$current_length\n";
        if($branchlength==1)      {
                if($tree_copy =~ /[\,\(][^\)\,\(]+:[^\,]/){}else{$x=0};
                                }else{
                if($tree_copy =~ /[\,\(][^\(\)\,]+[\,\)]/){}else{$x=0};
                                }
        };


$number_of_tips = $counter;
print "\nnumber_of_tips:$number_of_tips\n";


# now find each bifurcation ( seq_id:branchlength , seq_id:branchlength )
# record child nodes along with their lengths (if present)
```

```
# then remove bifurcation from string


print "creating tree structure.\n";


$x=1;
while ($x == 1)
        {
        $counter++;
        $node = $counter;
        if($branchlength==1)
                {
                $tree =~ s/\(([^:\(]*):([\d\.e\-]+)\,([^:]*):([\d\.e\-]+)\):([\d\.e\-
]+)/$counter:$5/;
                $child1 = $1;$child2 = $3;$node = $counter;$child1_bl = $2;$child2_bl = $4;$node_bl =
$5;


                $nodes{$node}{child1_bl} = $child1_bl;$nodes{$node}{child2_bl} = $child2_bl;
                $nodes{$node}{node_bl} = $node_bl;
                $nodes{$node}{sum_bl} = $nodes{$child1}{sum_bl} + $nodes{$child2}{sum_bl};
                }else{
                $tree =~ s/\(([^\,\(\)]+)\,([^\(\)\,]+)\)/$counter/; # replace bifurcation with node
number
                $child1 = $1;$child2 = $2;$node = $counter;
                }

        $nodes{$node}{is_node_a_tip}      = 0;
        $nodes{$node}{child1} = $child1;$nodes{$node}{child2} = $child2;
        $nodes{$child1}{parent} = $node;$nodes{$child2}{parent} = $node;
        $nodes{$child1}{which_child} = 1;$nodes{$child2}{which_child} = 2;

#         print "node:$node child1:$child1 child2:$child2\n";

        if($branchlength==1)
                {
                if($tree =~ /\([^:\(]*:[\d\.e\-]+\,[^:]*:[\d\.e\-]+\):[\d\.e\-]+/){}else{$x=0};
                }else{
                if($tree =~ /\(([^\,\(\)]+)\,([^\(\)\,]+)\)/){}else{$x=0};
                }

        };      # for each bifurcation

#######################################

        if($tree =~ /\([^\,\(\)]+\,[^\,\(\)]+\,[^\,\(\)]+\)/)
                {
                $root_trifurcation=1;
                print "root_trifurcation\n";
                ########################
                # similar to above but 3 instead of 2
                $counter++;
                $node = $counter;

                if($branchlength==1)
                        {
                        $tree =~ s/\(([^:\(]*):([\d\.e\-]+)\,([^:\(]*):([\d\.e\-
]+)\,([^:]*):([\d\.e\-]+)\)/$counter/; # appears no node_bl for root node:([\d\.e\-]+)
                        $child1 = $1;$child2 = $3;$child3 = $5;$node = $counter;$child1_bl =
$2;$child2_bl = $4;$child3_bl = $6; # $node_bl = $7;
                        $nodes{$node}{child1_bl} = $child1_bl;$nodes{$node}{child2_bl} =
$child2_bl;$nodes{$node}{child3_bl} = $child3_bl;
                        $nodes{$node}{sum_bl} = $nodes{$child1}{sum_bl} + $nodes{$child2}{sum_bl};
                        }else{
                        $tree =~ s/\(([^\,\(\)]+)\,([^\,\(\)]+)\,([^\(\)\,]+)\)/$counter/; # UNTESTED
                        $child1 = $1;$child2 = $2;$child3 = $3;$node = $counter;
                        }

                $nodes{$node}{is_node_a_tip}      = 0;
                $nodes{$node}{child1} = $child1;$nodes{$node}{child2} = $child2;$nodes{$node}{child3}
= $child3;
                $nodes{$child1}{parent} = $node;$nodes{$child2}{parent} =
$node;$nodes{$child3}{parent} = $node;
                $nodes{$child1}{which_child} = 1;$nodes{$child2}{which_child} =
2;$nodes{$child3}{which_child} = 3;

                ########################
                }else{print "NO_root_trifurcation\n";$root_trifurcation=0}
```

```
if($root_tree == 1)
        {
        if($root_trifurcation==1){die "\nwhy is there a root trifurcation with rooted tree?\n"};

        if($branchlength==1)
                {

                unless($tree_type == 5)
                {
                # appears necessary due to lack of brachlength for root node in rooted trees
                $counter++;
                $node = $counter;
                if($tree =~ /\([^:\(]*:[\d\.e\-]+\,[^:]*:[\d\.e\-]+\)/){}else{
                        print "BUG. tree:$tree . appears to be rooted tree with branchlengths.";
                        die " expected 2 child nodes left with no parent node branchlength but its
not here\n"}
                $tree =~ s/\(([^:\(]*):([\d\.e\-]+)\,([^:]*):([\d\.e\-]+)\)/$counter/; #
                $child1 = $1;$child2 = $3;$node = $counter;$child1_bl = $2;$child2_bl = $4;$node_bl =
0;
                $nodes{$node}{child1_bl} = $child1_bl;$nodes{$node}{child2_bl} = $child2_bl;
                $nodes{$node}{node_bl} = $node_bl;
                $nodes{$node}{sum_bl} = $nodes{$child1}{sum_bl} + $nodes{$child2}{sum_bl};
                }
                }else{
                # im pretty sure this would be taken care of
                }

        $nodes{$node}{is_node_a_tip}      = 0;
        $nodes{$node}{child1} = $child1;$nodes{$node}{child2} = $child2;
        $nodes{$child1}{parent} = $node;$nodes{$child2}{parent} = $node;
        $nodes{$child1}{which_child} = 1;$nodes{$child2}{which_child} = 2;

        }

$root_identity = $counter;

        print "\nstump! (should be a single node):$tree\n\nbranchlength:$branchlength
root_trifurcation:$root_trifurcation root_identity:$root_identity\n";



my $testingA;my $testingB;

my $subprune_count_total  =0;
my $subprune_count_list   =0;
my $subprune_count_unidentified =0;
my $subprune_count_pruned         =0;
my $subprune_count_notpruned      =0;

my @pre_pruning_nodekeys = keys %nodes;@pre_pruning_nodekeys = sort @pre_pruning_nodekeys;


foreach my $node(@pre_pruning_nodekeys) # go through each node
        {if($nodes{$node}{is_node_a_tip} == 1){    # if it is a tip

        $subprune_count_total++;
        my $cousin_node = "";


                if(exists($list_of_taxa_which_wont_be_pruned_from_added_trees2{$node}))
                        {
                        $subprune_count_list++;$testingA = 0;
                        }else{$testingA = 1}


        if($calculate_monophyly_scores_on_UNpruned_full_tree == 1){$testingA=0}

        if($testingA == 1 || $testingB == 1)
                {
                $subprune_count_pruned++;
                my $parent_id = $nodes{$node}{parent};my $grandparent_id =
$nodes{$parent_id}{parent};my $add_bl=0;
                if ($nodes{$node}{which_child} == 1)
                        {               # find cousin node
                        $cousin_node = $nodes{$parent_id}{child2};$add_bl =
$nodes{$parent_id}{child2_bl}
                        # print $nodes{$node}{which_child};print " = child1\n";
                        }else{
                        $cousin_node = $nodes{$parent_id}{child1};$add_bl =
$nodes{$parent_id}{child1_bl}
                        # print "$node " , $nodes{$node}{which_child};print " = child2\n";
```

```
                        }

                unless($node =~ /^$root_identity$/ || $parent_id =~ /^$root_identity$/) # HACK: dont
delete unidentified if node or parent is the root
                        {                                                                       # pain
in the ass to reset the root

                        if ($nodes{$parent_id}{which_child} == 1)
                                {       # parent node is no longer required, so link cousin node to
grandparent node
                                $nodes{$grandparent_id}{child1} = $cousin_node;     # and grandparent
to cousin
                                $nodes{$cousin_node}{parent} = $grandparent_id;
                                $nodes{$cousin_node}{which_child} = 1;

                        #       print "cousin_node:$cousin_node ";
                        #       print "nodes{grandparent_id}{child1_bl}:",
$nodes{$grandparent_id}{child1_bl};
                        #       print " nodes{cousin_node}{node_bl}:",
$nodes{$cousin_node}{node_bl};
                        #       print "add_bl:$add_bl\n";die;

                                $nodes{$grandparent_id}{child1_bl} =
$nodes{$grandparent_id}{child1_bl} + $add_bl; # cousin may not have node_bl if it is a tip
                                $nodes{$cousin_node}{node_bl} = $nodes{$grandparent_id}{child1_bl};
                                }else{
                                $nodes{$grandparent_id}{child2} = $cousin_node;
                                $nodes{$cousin_node}{parent} = $grandparent_id;
                                $nodes{$cousin_node}{which_child} = 2;
                                $nodes{$grandparent_id}{child2_bl} =
$nodes{$grandparent_id}{child2_bl} + $add_bl;
                                $nodes{$cousin_node}{node_bl} = $nodes{$grandparent_id}{child2_bl};
                                }

        #               print "tip:$node is unidentified. deleting it and parent:$parent_id\n";
        #               print "node:$node has no identified genera. $node is:";     print
$nodes{$node}{which_child};
        #               print " of parent:$parent_id . deleting and linking parent (child:";print
$nodes{$parent_id}{which_child};
        #               print " of $grandparent_id) to cousin node:$cousin_node\n";

                        # finally delete node and parent node
                        delete($nodes{$node});delete($nodes{$parent_id});

                        }

                }else{
                $subprune_count_notpruned++;
                }


        }}
#open (RESULTS_OUT, ">>wasp_rarefaction_logfile") || die "cant open log file\n";
#print RESULTS_OUT "subprune_count_total:$subprune_count_total
subprune_count_list:$subprune_count_list subprune_count_unidentified:$subprune_count_unidentified ";
#print RESULTS_OUT "subprune_count_pruned$subprune_count_pruned
subprune_count_notpruned:$subprune_count_notpruned\n";
#close(RESULTS_OUT);



$id = $root_identity;
$root_bl = $nodes{$id}{node_bl};
$newick_string= "($id)";
# $newick_string_bl= "($id)";
$newick_string_bl= "($id:$root_bl)";

print "\ngenerating newick string from tree structure\n";

############################
generate_newick_string($id);#
############################

        $newick_string_bl =~ s/^\(((\(.+\))[^\)]+\)$/$1/;
        ###################
        $newick_string_bl =~ s/([^\;])$/$1\;/;              # add semi-colon



#print "newick_string_bl:$newick_string_bl\n";
```

```
return($newick_string_bl);


sub generate_newick_string
        {
        # note identified here may refer to identified tip or internal node. unidentified is tip only
        my $next = shift;#       my $prev = $next;
        my @next1 = ();
        $next1[0] = $nodes{$next}{child1};$next1[1] = $nodes{$next}{child2};
        if($root_trifurcation == 1 && $next =~ /$root_identity/)
                {
                $next1[2] = $nodes{$next}{child3};
        #       print "root_trifurcation next:$next nodes{next}{child1}:" , $nodes{$next}{child1} , "
nodes{next}{child2}:" , $nodes{$next}{child2} , " nodes{next}{child3}:" , $nodes{$next}{child3} ,"\n";
                }else{
        #        print "next:$next nodes{next}{child1}:" , $nodes{$next}{child1} , "
nodes{next}{child2}:" , $nodes{$next}{child2} , "\n";
                }

        if($next1[0] =~ /^[^_]+_[^_]+_$question_mark_replacement_string/ || $next1[1] =~
/^[^_]+_[^_]+_$question_mark_replacement_string/)
                {       # BRAC_HELCONINAE_?_?
                }

        if($root_trifurcation == 1 && $next =~ /$root_identity/)
                {
                $newick_string     =~
s/([\,\(\)])$next([\,\(\)\:])/$1\($nodes{$next}{child1}\,$nodes{$next}{child2}\,$nodes{$next}{child3}\
)$2/;
                $newick_string_bl =~
s/([\,\(\)])$next([\,\(\)\:])/$1\($nodes{$next}{child1}\:$nodes{$next}{child1_bl}\,$nodes{$next}{child
2}\:$nodes{$next}{child2_bl}\,$nodes{$next}{child3}\:$nodes{$next}{child3_bl}\)$2/;
                }else{
                $newick_string     =~
s/([\,\(\)])$next([\,\(\)\:])/$1\($nodes{$next}{child1}\,$nodes{$next}{child2}\)$2/;
                $newick_string_bl =~
s/([\,\(\)])$next([\,\(\)\:])/$1\($nodes{$next}{child1}\:$nodes{$next}{child1_bl}\,$nodes{$next}{child
2}\:$nodes{$next}{child2_bl}\)$2/;
                }

#         print "newick_string:$newick_string\n";
#         print "newick_string_bl:$newick_string_bl\n";

        for my $index(0 .. $#next1)
                {
                $test = @next1[$index];
                # if($nodes{$test}{is_node_a_tip} == 0 ){generate_newick_string($test)} # disabled
due to id's at root trifurcation
                if($test=~ /^\d+$/ ){generate_newick_string($test)} # RECURSION, if non-tip
                }
        return();
        }
        }
```

Appendix 1.4: calculate_taxonomic_ri. Reads tree and taxon key files, then makes appropriate paup input file to calculate the tRI of the contained groups (chapters two and three).

```
sub calculate_taxonomic_ri
        {


my $sub_input = shift;

#print "sub_input:$sub_input\n";

my $ensemble;

my @sub_input_array = split(/__/, $sub_input);

my $sia0 = $sub_input_array[0];

if(length($sia0)<=0){die "
        ERROR. sub calculate_taxonomic_ri has been called but no tax groups specified.
        this may happen if you have specified the incorrect keyfile. quitting
        "}
#print "sia0:$sia0\n";

my @test_this_group_array = split(/ /,$sia0);
#$test_this_group

if(scalar @test_this_group_array >= 3)
        {
        #print "sub calculate_taxonomic_ri. looks like ENSEMBLE\ntaxa in tree:$number_taxa\n";
        $ensemble=1
        }else{
        # print "sub calculate_taxonomic_ri. looks like single group :@test_this_group_array:\n";
        $ensemble=0}


my $currenttree = $sub_input_array[1];
my $tree = $currenttree;
my %ids = ();

$tree =~ s/\s//g;

# for raxml usually get fp numbers, occasionaly get this: 1.0812708e-05 so .............
$tree =~ s/:\d\.\d+e\-\d+([\,\(\)])/$1/g;

# regular raxml fp node labels
$tree =~ s/:\d+\.\d+([\,\(\)])/$1/g;

# paup parsimony node labels:
$tree =~ s/:\d+([\,\(\)])/$1/g;


#print "$tree\n";


while($tree =~ /[\(\)\,][^\(\)\,]+[\(\)\,]/)
        {
        $tree =~ s/([\(\)\,])([^\(\)\,]+)([\(\)\,])/$1$3/;
        my $tha_taxon = $2;
        $ids{$tha_taxon} =1;
        if($tha_taxon =~ /\-/)
                {
                print "$tha_taxon\n";die "BUG\n"
                }
        }

my @ids_in_sub_tree = keys %ids;@ids_in_sub_tree = sort @ids_in_sub_tree;


my $debug_current_tax="";

my $number_taxa = scalar @ids_in_sub_tree;
#print "taxa in tree:$number_taxa\n";


open (OUT , ">tax_ri_paup_input") || die "no openage\n";
print OUT "\#NEXUS\nBEGIN TAXA\;\nDIMENSIONS NTAX=$number_taxa\;TAXLABELS\n";
```

```
foreach my $tax(@ids_in_sub_tree)
        {
        print OUT "\t$tax\n";
        };

print OUT "\;\nENDBLOCK\;\n\nBEGIN CHARACTERS\;\nDIMENSIONS NCHAR=" , scalar @test_this_group_array ,
"\;\nFORMAT DATATYPE=STANDARD GAP=- SYMBOLS=\"01\"\;
CHARLABELS\n";


my $char_counter = 1;
my $sf = $test_this_group;
#unless($sf=~ /$question_mark_replacement_string|^[a-z]+$|\d/){print OUT " \[$char_counter\]
$sf\n";$char_counter++};

print OUT "\;\n  MATRIX\n";

foreach my $tax(@ids_in_sub_tree)
        {
        print OUT "\t$tax ";

        foreach my $test_this_group(@test_this_group_array)
                {

#               if($ensemble ==1) {
#                                   $current_rank_being_tested =
$what_rank_is_this_taxon{$test_this_group};
#                                   unless(exists($what_rank_is_this_taxon{$test_this_group})){die "why
no rank allocation?\n"}
#                                   }

                if($current_rank_being_tested eq "genus" )
                        {
                        if($tax_relationships{$tax}{genus} =~ /^$test_this_group$/)
                                {
                                $debug_current_tax = $tax;print OUT "1";#print "tax:$tax belongs to
genus $test_this_group\n";
                                }else{print OUT "0"}
                        }
                if($current_rank_being_tested eq "subfamily")
                        {
                        if($tax_relationships{$tax}{subfamily} =~
/^$test_this_group$/){$debug_current_tax = $tax;print OUT "1"}else{print OUT "0"}
                        }
                if($current_rank_being_tested eq "family" )
                        {
                        if($tax_relationships{$tax}{family} =~
/^$test_this_group$/){$debug_current_tax = $tax;print OUT "1"}else{print OUT "0"}
                        }
                }

        print OUT "\n";
        };

print OUT "\;\nENDBLOCK\;\nBEGIN TREES\;\ntree tree1 = $currenttree\n\;\nENDBLOCK\;\nBEGIN PAUP\;\nset
autoclose=yes\;\nPSCORES 1 /CI=YES RI=YES\;\nQUIT\;\nENDBLOCK\;\n";

close(OUT);

my $ri=""; my $ci="";

$paup_screenout = "paup_screenout";
system("rm $paup_screenout");
system("paup tax_ri_paup_input -n > $paup_screenout");



open(IN4, $paup_screenout) || die "cant open paup_screenout $paup_screenout\n";
while ($line = <IN4>)
        {if ($line =~ /^RI\s+(\S+)$/){$ri = $1};
        if ($line =~ /^CI\s+(\S+)$/){$ci = $1}};
close(IN4);

if(length($ri)<=0){die "cant read RI from paup file\n"}


if ($ri =~ /0\/0/){$ri = ""}

my $print_this_before_score = "";
if($ensemble ==1)
        {
```

```
        $print_this_before_score="ensemble";

        print "sub calculate_taxonomic_ri. looks like ENSEMBLE ri:$ri\ntaxa in tree:$number_taxa\n";

        }else{
        $print_this_before_score = $sia0;
if ($ri =~ /^\d\.\d+$/)
        {
        print RESULTS_OUT "$print_this_before_score:$ri\t";
        }else{
        print RESULTS_OUT "$print_this_before_score:NA\t";
        };


        }

return($ri);

}
```

Appendix 1.5: raxml wrapper. Accounts for format conversion and incompatable sequence id's. Used for analyses in chapters two and three.

```
sub build_ml_tree
{

my $current_resampled_alignment = shift;

my $parsimony_tree = "";
my $parsimony_tree_with_branchlengths = "";
my $ml_tree = "";
my $current_number;

my $current_seed = localtime time; $current_seed=~ s/\D//g;

my $which_version_of_raxml = 2; # 1=older version 2=raxmlHPC-7.2.5
print "raxml_tree on $current_resampled_alignment.\n";

open (IN, $current_resampled_alignment) || die "cant open infile $current_resampled_alignment\n";
open (OUT, ">$current_resampled_alignment.recoded") || die "cant open outfile\n";

print "infile ($infile) sucessfully read. changing taxoncodes to code of suitable length.\n";

my $s1_entry_counter=0;
my @key_array = ();
while (my $line=<IN>)
        {
        $line =~ s/\n//;$line =~ s/\r//;
        if ($line =~ /^>(\S+)$/ )
                {
                my $id = $1;my $seq = $2;my $current_number;
                if(length($s1_entry_counter) == 1){$current_number = "0000" . $s1_entry_counter};
                if(length($s1_entry_counter) == 2){$current_number = "000" . $s1_entry_counter};
                if(length($s1_entry_counter) == 3){$current_number = "00" . $s1_entry_counter};
                if(length($s1_entry_counter) == 4){$current_number = "0" . $s1_entry_counter};

                if($id =~ /outgroup/i){$current_number = "outgroup1"}

                print OUT ">D$current_number\n";
                $key_array[$s1_entry_counter] = "$current_number $id";
                $s1_entry_counter++;
                }else{print OUT "$line\n"}; #
        };
close(IN);
close(OUT);



my $phylip_alignment = $current_resampled_alignment . ".phy";
system("rm $phylip_alignment");

$format_conversion_input_file = $current_resampled_alignment . ".recoded";
$format_conversion_output_file = $phylip_alignment;

##########################
fasta_format_to_phylip();#
##########################

system("rm *.raxml_output");              # -y will only print parsimony tree



my $raxml_tree_file;

############################

my $raxml_command;

        $raxml_command = "raxmlHPC_7_2_5 -D"; # -D switch stops treesearch at asymptopic phase
        $raxml_tree_file = "RAxML_bestTree.raxml_output";

$command = $raxml_command . " -s $phylip_alignment -n raxml_output -p $current_seed -m GTRCAT -c 4 -o
Doutgroup1";

# $command = $raxml_command . " -s $phylip_alignment -n $nexus_alignment.raxml_output -m GTRCAT -c 4 -
q part_file"; # -q may26_Qmissingchar_1of2_trimmedCOI_4gene.raxml_partition_file
#  -q raxml_partition_file -o D00000");
```

```
                # $command = "raxml-pthreads -T 10 -m GTRCAT -s $phylip_alignment -n
$nexus_alignment.raxml_output";

        print "command:\n$command\n";
        system("$command"); #### RUN RAXML ####



open(TREEFILE, $raxml_tree_file) || die "cant open treefile\n";
while (my $line = <TREEFILE>)
        {$line=~ s/\n//;$line=~ s/\r//;
        $ml_tree .= $line};
close(TREEFILE);


my $line_counter=0;
foreach my $line(@key_array)
        {
        my @split_line = split(/ /,$line);
        my $find_this = $split_line[0];
        $ml_tree =~ s/D$find_this(\D)/$split_line[1]$1/;

        $line_counter++;
        };

# print "ml_tree::$ml_tree\n";

my $tree_bytes = length($ml_tree);print "tree is length $tree_bytes bytes\n";

@key_array = (); # just in case!
####################################system("rm *.raxml_output");

$ml_tree =~ s/\s//g;

if($ml_tree =~ /D\d\d\d\d\d[^a-zA-Z0-9]/)
        {
        print "\n$ml_tree\n";
        die "NON reassignment of terminal label. sub build_ml_tree\n";
        }



open(LOG, ">>locus_contribution_LOG") || die "cant open logfile\n";
print LOG "parsimony_tree_with_branchlengths:$current_resampled_alignment:\t$ml_tree\n";
close(LOG);

open(RES, ">>current_analysis_log") || die "cant open file\n";
print RES "is_main_locus:$is_current_tree_main_locus\ttree:$ml_tree\n";
$current_analysis_log_string .= "is_main_locus:$is_current_tree_main_locus\ttree:$ml_tree\n";
close(RES);

system("rm $format_conversion_input_file");system("rm $format_conversion_output_file");

}

sub fasta_format_to_phylip
        {

        my $current_seq_length;
        my $missing_data_character = "N";
        my $current_id;
        my $current_seq_length;
        my %fasta_seqs = ();

        open(FASTA_IN, $format_conversion_input_file) || die "Cant open
input:$format_conversion_input_file.\n";
        my $file_as_string = ""; my @all_lines = ();
        while($line= <FASTA_IN>){$file_as_string .= $line};
        close(FASTA_IN);

        $file_as_string =~ s/\012\015?|\015\012?/\n/g;
        @all_lines = split />/, $file_as_string;


        for my $each_line(1 .. $#all_lines)
                {
                my $line = $all_lines[$each_line];
                        # following 2 lines untested in current context

                if ($line =~ /^(.+)\n/ )
                        {
```

```
                        $current_id = $1;$line =~ s/^.+\n//;       $line =~ s/\n//g;$line =~
s/\r//g;$current_seq_length = length($line);
                        $line =~ s/(.)$/$1\n/;
                        }else{die "BUG\n"};

        #       print "current_id:$current_id\ncurrent_sequence:$line\n";
                $fasta_seqs{$current_id} = $line;
                }

        my @fasta_seqs_keys = keys %fasta_seqs;

        my $number_of_taxa = scalar @fasta_seqs_keys;
        print  "$number_of_taxa entries read into memory. length:$current_seq_length\n";


        open(PHYLIP_OUT ,">$format_conversion_output_file") || die "cant open output
file:$format_conversion_output_file\n";

        print PHYLIP_OUT "$number_of_taxa $current_seq_length\n";

        for my $i(0 .. $#fasta_seqs_keys)
                {
                my $current_name = $fasta_seqs_keys[$i];
                my $current_seq = $fasta_seqs{$current_name};

                if(length($current_name)<=1 || length($current_seq)<=1){print "warning: zero length
of current entry. quitting\n";die}

                if($current_name =~ /\S\s\S|\S-\S|\S\(\S/) # |\S-\S
                        {
                        die "not expecting spaces in names\n";#print NEXUS_OUT "\'$current_name\'
$current_seq\n";
                        }else{

                        if(length($current_name)==3){print PHYLIP_OUT "$current_name
$current_seq\n"}
                        if(length($current_name)==4){print PHYLIP_OUT "$current_name
$current_seq\n"}
                        if(length($current_name)==5){print PHYLIP_OUT "$current_name
$current_seq\n"}
                        if(length($current_name)==6){print PHYLIP_OUT "$current_name
$current_seq\n"}
                        if(length($current_name)==7){print PHYLIP_OUT "$current_name
$current_seq\n"}
                        if(length($current_name)==8){print PHYLIP_OUT "$current_name
$current_seq\n"}
                        if(length($current_name)==9){print PHYLIP_OUT "$current_name $current_seq\n"}

                        if(length($current_name)>=10){print PHYLIP_OUT "$current_name
$current_seq\n"}

                        #print PHYLIP_OUT "$current_name $current_seq\n";



                        }
                };

        close(PHYLIP_OUT);

        };
```

Appendix 1.6: Geographic website scraping subroutines. Requires the Perl module www:mechanize.

```
sub obtain_species_from_zipcodezoo # get list of species belonging to current genus from zipcodezoo
        {
                # http://zipcodezoo.com/Key/Animalia/Lucanus_Genus.asp
        my $subroutine_genus = shift;
        my $current_website = "http://zipcodezoo.com/Key/Animalia/" . $subroutine_genus .
"_Genus.asp";

        #print "\nzcz_genus_website:$current_website\n";
        my $mech = WWW::Mechanize->new();
        $mech->get($current_website);
        $screenout = $mech->content;
        my $are_species_present = 0;
        if ($screenout =~ /The page cannot be found/)
                {
                print "zcz_genus_website NOT FOUND\n";
                print "N\t";
                }else{
                                # middo
                while ($screenout =~ /href\W{1,5}Animals\/\w\/$subroutine_genus[_](\w+)\W/i)
                        {
                        my $found_species = $1;
                        $zcz_complete_species_hash{$subroutine_genus} .= $found_species . " ";
        # append each into hash entry for genus. 2hashs used
                        unless($complete_species_hash{$subroutine_genus}=~ /^$found_species\s/ ||
$complete_species_hash{$subroutine_genus}=~
/\S$found_species\s/){$complete_species_hash{$subroutine_genus} .= $found_species . " "};
                        $screenout =~
s/href\W{1,5}Animals\/\w\/$subroutine_genus[_]$found_species\W//i;
                        $are_species_present++;
                        };

                };
        print "$are_species_present species for $subroutine_genus on zipcodezoo\n";
        return(); # %zcz_complete_species_hash

        } # sub obtain_species_from_zipcodezoo



sub obtain_species_from_unep
        {
        my $subroutine_genus = shift;
        #######$genus = "lucanus";$current_species = "cervus";
        ####$genus = "cicindela";

        my $unep_genus_website = "http://www.unep-wcmc.org/isdb/Taxonomy/tax-genus-
search1.cfm?displaylanguage=ENG&source=animals&habit=";
        my $mech = WWW::Mechanize->new();

        my $form_found=0;my $form_attempt=0;
        while($form_found==0)
                {
                $form_attempt++;print "unep form_attempt:$form_attempt\n";
                $mech->get($unep_genus_website);
                unless($mech->content=~/encountered an internal error and was unable to complete your
request/ || length($mech->content)<=50){$form_found=1}
                }
        open(OUT_DEBUG, ">debug_screenout") || die "cant open out";
        print OUT_DEBUG $mech->content;
        close(OUT_DEBUG);

        $mech->submit_form        ( fields    =>    { GName  => $subroutine_genus  } );
        $mech->click();

        my $screenout = $mech->content;
        my $are_species_present = 0;
        if ($screenout =~ /The page cannot be found/)
                {
                print "unep_genus_website NOT FOUND\n";die;
                print "N\t";
                }else{
                my @all_lines2 = split /\012\015?|\015\012?/, $screenout; $screenout = join(//,
@all_lines2);

                while ($screenout =~ /genus=$subroutine_genus[^\;]+\;species=(\w+)\W/i)
```

```
                        {
                        my $found_species = $1;
                        $unep_complete_species_hash{$subroutine_genus} .= $found_species . " ";
                        unless($complete_species_hash{$subroutine_genus}=~ /^$found_species\s/ ||
$complete_species_hash{$subroutine_genus}=~
/\W$found_species\s/){$complete_species_hash{$subroutine_genus} .= $found_species . " "};
                        $screenout =~ s/genus=$subroutine_genus[^\;]+\;species=\w+\W//i;
                        $are_species_present++;
                        };


                }

        print "$are_species_present species for $subroutine_genus on UNEP\n";
        return();

        };


sub obtain_species_from_fe
        {
        my $subroutine_genus = shift;
        ### $subroutine_genus = "lucanus";$current_species = "cervus";
        ### $subroutine_genus = "cicindela";
        my $mech = WWW::Mechanize->new();
        $mech->get($fauna_website);
        $screenout = $mech->content;
        $mech->submit_form        ( fields => {genus1  => $subroutine_genus } , button    => 'submit'
);
        $screenout = $mech->content;

        my $are_species_present = 0;
        if ($screenout =~ /The page cannot be found/)
                {
                print "col_genus_website NOT FOUND\n";die;
                }else{
                my @all_lines2 = split /\012\015?|\015\012?/, $screenout; $screenout = join(//,
@all_lines2);

                # span class="search_results"><i>Acanthoscelides plagiatus<
                # span class="search_results"><i>Cicindela (Cicindela) monticola albanica
                # sulcatus semisulcatus
                $screenout =~ s/\(\w+\)//g;        # rm brackets
                while ($screenout =~ /span
class.{1,3}search_results.{5,8}$subroutine_genus\s{1,2}(\w+)\W/i)
                        {
                        my $found_species = $1; # print "found_species:$found_species\n";
                        unless($fe_complete_species_hash{$subroutine_genus}=~ /\W$found_species\s/ ||
$fe_complete_species_hash{$subroutine_genus}=~ /^$found_species\s/)
                                {$fe_complete_species_hash{$subroutine_genus} .= $found_species . "
"};

                        unless($complete_species_hash{$subroutine_genus}=~ /\W$found_species\s/ ||
$complete_species_hash{$subroutine_genus}=~ /^$found_species\s/)
                                {$complete_species_hash{$subroutine_genus} .= $found_species . " "};


                        $screenout =~ s/span
class.{1,3}search_results.{5,8}$subroutine_genus\s{1,2}\w+\W//i;
                        $are_species_present++;
                        };
                }

        print "$are_species_present species for $subroutine_genus on FE\n"; # ,
$fe_complete_species_hash{$subroutine_genus};

        return();
        };

sub scrape_zipcodezoo
        {
        my $subroutine_binomial = shift;
        my $subroutine_genus;my $subroutine_species;
        if($subroutine_binomial =~
/^(\w+)[_](\w+)$/){$subroutine_genus=$1;$subroutine_species=$2}else{die "\nBUG!\n"}


        my $found_correct_adress=0;        # i have written it so it searches for variations in the
web adress for species
                                        # im not certain this is neccesary, but it appeared to be
at the time
```

```
        my  $current_website = "http://zipcodezoo.com/animals/" . substr($subroutine_species,0,1) .
"/" .
        $subroutine_genus . "_" . $subroutine_species  . "/Distribution.asp";
        my $mech = WWW::Mechanize->new();
        $mech->get($current_website);
        $screenout = $mech->content;
        unless($screenout =~ /The page cannot be found/){$found_correct_adress=1};


        if($found_correct_adress==0)
        {
        my  $current_website = "http://zipcodezoo.com/animals/" . substr($subroutine_genus,0,1) . "/"
.
        $subroutine_genus . "_" . $subroutine_species  . "/Distribution.asp";
        $mech->get($current_website);
        $screenout = $mech->content;
        unless($screenout =~ /The page cannot be found/){$found_correct_adress=1};
        };


        if($found_correct_adress==0)
        {
        my $copy_subroutine_genus = $subroutine_genus;      # some have lowercase genus character in
adress
        $copy_subroutine_genus =~ tr/[A-Z]/[a-z]/;
        my  $current_website = "http://zipcodezoo.com/animals/" . substr($subroutine_genus,0,1) . "/"
.
        $copy_subroutine_genus . "_" . $subroutine_species  . "/Distribution.asp";
        $mech->get($current_website);
        $screenout = $mech->content;
        unless($screenout =~ /The page cannot be found/){$found_correct_adress=1};
        }

        if($found_correct_adress==0)
        {
        my  $current_website = "http://zipcodezoo.com/Animals/" . substr($subroutine_genus,0,1) . "/"
.
        $subroutine_genus . "_" . $subroutine_species  . "/Distribution.asp";
        $mech->get($current_website);
        $screenout = $mech->content;
        unless($screenout =~ /The page cannot be found/){$found_correct_adress=1};
        };


        if($found_correct_adress==1){scrape_zipcodezoo_part2()}


        }


sub scrape_zipcodezoo_part2
        {

        # lin_list::</h2><div class="political">   <img
src="/Animals/A/Acanthoscelides_brevipes/Maps/Countries.jpg">       <p>
        Costa Rica &#149; Venezuela       </p></div><h2>Observations</h2><a
id="observationsmap"></a><div class="backtotop">[ <a href="#top">Back to top</a> ]</div><div
id="observationsdiv" class="slide"><div id="gmapsdiv"><iframe
src="/Animals/A/Acanthoscelides_brevipes/Maps/GMap.asp" frameborder="no"></iframe><small><p><a
href="http://zipcodezoo.com/about/DataSources.asp">
        # if($subroutine_species =~ /brevipes/){print "lin_list:$screenout";die;}

        my @all_lines2 = split /\012\015?|\015\012?/, $screenout; $screenout = join(//, @all_lines2);
        # $screenout =~ s/\n//g;$screenout =~ s/\r//g;$screenout =~ s/\f//g;$screenout =~ s/\e//g;

        $screenout =~ s/^.+Political Regions//;
        $screenout =~ s/Observations.+$//g;
        $screenout =~  s/<[^>]+>//g;
        $screenout =~ s/Range and Population//g;
        $screenout =~ s/Countries\://g;
        $screenout =~ s/\t//g;
        $screenout =~  s/  / /g;
        $screenout =~ s/\&\#149\;//g;
        $screenout =~ s/\(Ref. \d+\)//g;

        if($screenout =~ /\w{2,30}/)
                {
                open(OUT, ">>distribution_list") || die "cant open outfileE\n";print OUT
"**zipcodezoo** $screenout ";close(OUT);
```

```
                ###################################################################

                }else{
                print "no_countries_for_ZCZ\n";
                }
        # my $html = $lin_list;
        # look_for_countries_in_html($html);
        }



sub scrape_unep
        {
        my $subroutine_binomial = shift;
        my $subroutine_genus;my $subroutine_species;
        if($subroutine_binomial =~
/^(\w+)[_](\w+)$/){$subroutine_genus=$1;$subroutine_species=$2}else{die "\nBUG!\n"}


        #######$subroutine_genus = "lucanus";$subroutine_species = "cervus";
        #######$genus = "cicindela";
        my $unep_website = "http://www.unep-wcmc.org/isdb/Taxonomy/tax-gs-
search1.cfm?displaylanguage=ENG&source=animals&habit=";
        my $mech = WWW::Mechanize->new();
#       $mech->get($unep_website);

        my $form_found=0;my $form_attempt=0;
        while($form_found==0)
                {
                $form_attempt++;print "unep form_attempt:$form_attempt\n";
                $mech->get($unep_website);
                unless($mech->content=~/encountered an internal error and was unable to complete your
request/ || length($mech->content)<=50){$form_found=1}
                }
        open(OUT_DEBUG, ">debug_screenout") || die "cant open out";
        print OUT_DEBUG $mech->content;
        close(OUT_DEBUG);


        $mech->submit_form        ( fields =>
                {GenName         => $subroutine_genus ,
                SpcName  => $subroutine_species } );
        $mech->click();


        my $screenout = $mech->content;

        if($screenout =~/href=\"(\/isdb\/Taxonomy\/tax.*distribution)\"\>/ )
                {
                print "unep regex_found\n";
                print OUT3 "unep regex_found\n";

                my $next_website = "http://www.unep-wcmc.org" . $1;

                # print "going to:\n$next_website\n";
                $mech->get($next_website);
                $screenout = $mech->content;
                # @thing = split(/\n/,$screenout);

                my @all_lines2 = split /\012\015?|\015\012?/, $screenout;

                my $in_dist=0; my $lin_list="";
                foreach $dist_line(@all_lines2)
                        {
                        $dist_line =~ s/\n//;$dist_line =~ s/\r//;
                        if ($dist_line=~/^\W+Distribution\W+$/){$in_dist=1};
                        if ($dist_line=~/Distribution key\W+$/ && $in_dist == 1){$in_dist=0};
                        if ($dist_line=~/font face/ && $in_dist == 1){$in_dist=0};
                        if ($in_dist == 1){$lin_list = $lin_list . $dist_line};
                        };

                $lin_list =~  s/<[^>]+>//g;
                $lin_list =~  s/\t//g;
                $lin_list =~  s/  / /g;
                $lin_list =~  s/Distribution//g;

                ###################################################################

                open(OUT, ">>distribution_list") || die "cant open outfileF\n";print OUT "**unep**
$lin_list ";close(OUT);
```

```
                }else{
                print OUT3 "unep regex_not_found\n";
                };
        };

sub scrape_col
        {
        my $subroutine_binomial = shift;
        my $subroutine_genus;my $subroutine_species;
        if($subroutine_binomial =~
/^(\w+)[_](\w+)$/){$subroutine_genus=$1;$subroutine_species=$2}else{die "\nBUG!\n"}

        my $col_website = "http://www.catalogueoflife.org/search_scientific.php";
        my $mech = WWW::Mechanize->new();
        $mech->get($col_website);


        $mech->submit_form        (
                form_name => 'search_form',
                fields    =>     {genus  => $subroutine_genus , species => $subroutine_species} ,
                button    => 'Submit'
                                );

                        #looking for:    show_species_details.php?record_id=5161564'><i>Cicindela
abdominali
                        # translate into:
http://www.catalogueoflife.org/show_species_details.php?record_id=5161564
        $screenout = $mech->content;

        my $col_distribution_list = "";
        my @all_lines2 = split /\012\015?|\015\012?/, $screenout;
        foreach $dist_line(@all_lines2)
                {
                if ($dist_line=~/show_species_details.php.record_id.(\d+).+/)
                        {
                        # my $col_id = $1;
                        my $target_col_website =
"http://www.catalogueoflife.org/show_species_details.php?record_id=" . $1;
                        print "catalouge of life id:$1\n";
                        $mech->get($target_col_website);
                        $screenout = $mech->content;
                        my $in_dist=0;
                        @thing2 = split(/\n/,$screenout);
                        foreach $dist_line2(@thing2)
                                {
                                if($dist_line2=~/>Distribution:/i){$in_dist=1};
                                if($dist_line2=~/Additional data/ && $in_dist==1){$in_dist=0};
                                if ($in_dist == 1){$col_distribution_list = $col_distribution_list .
$dist_line2};
                                };

                        };

                };
                # $col_distribution_list=~s/<[-<>]+>//g;
                my $col_distribution_list2="";
                $col_distribution_list=~s/\s\s+//g;

                if ($col_distribution_list=~/Distribution.+fieldvalue(.+)</)
                        {
                        $col_distribution_list2 = $1;
                        };#else{open(OUT, ">>distribution_list") || die "cant open outfileG\n";print
OUT $col_distribution_list;close(OUT)};
                        ##########$col_distribution_list2=~s/<.+>//;
                        $col_distribution_list2 =~  s/<[^>]+>//g;
                        if($col_distribution_list2 =~ /\w\w\w/)
                                {

                                ######################################################

                                open(OUT, ">>distribution_list") || die "cant open outfileH\n";print
OUT "**col** $col_distribution_list2 ";close(OUT);

                                ######################################################

                                 ##### my $html = $col_distribution_list2;
look_for_countries_in_html($html);
                                };#if(length($col_distribution_list2)>=1)

        };
```

```
sub scrape_fe
        {
        my $subroutine_binomial = shift;
        my $subroutine_genus;my $subroutine_species;
        if($subroutine_binomial =~
/^(\w+)[_](\w+)$/){$subroutine_genus=$1;$subroutine_species=$2}else{die "\nBUG!\n"}

        ######$subroutine_genus = "lucanus";$subroutine_species = "cervus";
        my $mech = WWW::Mechanize->new();
        $mech->get($fauna_website);
        $screenout = $mech->content;

        $mech->submit_form        (
                                  fields => {      genus1  => $subroutine_genus ,
                                  species1 => $subroutine_species   } ,
                                  button    => 'submit'
                                  );

        $screenout = $mech->content;

        # looking for: name="show_1_0" value="123296" onClick


        my @all_lines2 = split /\012\015?|\015\012?/, $screenout;
        my $faunaeuropea_result=0;
        my $faunaeuropea_id="";
        foreach $dist_line(@all_lines2)
                {
                if ($dist_line=~/name.+show_1_0.+value..(\d{1,10})\" onClick/)
                        {
                        $faunaeuropea_id = $1;
                        $target_faunaeuropea_website = "http://www.faunaeur.org/full_results.php?id="
. $faunaeuropea_id;
                        print "faunaeuropea_id:$faunaeuropea_id\n";
                        print OUT3 "faunaeuropea_id:$faunaeuropea_id\n";
                        $faunaeuropea_result=1;
                        };
                };


        if($faunaeuropea_result==1)
                {
                my $faunaeuropea_test1=0;my $faunaeuropea_test2=0;
                # parse id from above then place in site below

                $mech->get("$target_faunaeuropea_website");
                $screenout = $mech->content;

                $mech->submit_form(form_number => 2);
                $mech->click();
                $screenout = $mech->content;# print $screenout;
                @all_lines2 = split /\012\015?|\015\012?/, $screenout;
                my $faunaeuropea_result=0; my $faunaeuropea_id="";
                my $faunaeuropea_list = "";

                foreach $dist_line(@all_lines2)
                        {
                        if($dist_line=~/($subroutine_genus)/i){$faunaeuropea_test1=1};
                        if($dist_line=~/($subroutine_species)/i){$faunaeuropea_test2=1};

                        if($dist_line=~/(Afro-tropical region)\t.*Presen/)  {$faunaeuropea_list .= "
" . $1};
                        if($dist_line=~/(Australian region)\t.*Presen/)
        {$faunaeuropea_list .= " " . $1};
                        if($dist_line=~/(East Palaearctic)\t.*Presen/)
        {$faunaeuropea_list .= " " . $1};
                        if($dist_line=~/(Near East)\t.*Presen/)
        {$faunaeuropea_list .= " " . $1};
                        if($dist_line=~/(Nearctic region)\t.*Presen/)
        {$faunaeuropea_list .= " " . $1};
                        if($dist_line=~/(Neotropical region)\t.*Presen/)    {$faunaeuropea_list .= "
" . $1};
                        if($dist_line=~/(North Africa)\t.*Presen/)          {$faunaeuropea_list .= "
" . $1};
                        if($dist_line=~/(Oriental region)\t.*Presen/)
        {$faunaeuropea_list .= " " . $1};

                        };
```

```
                if($faunaeuropea_test1==1 && $faunaeuropea_test2==1)
                        {
                        print OUT2 "**faunaeuropea** $faunaeuropea_list";
                        print OUT3 "**faunaeuropea_list** $faunaeuropea_list\n";


                        ######################################################################

                        open(OUT, ">>distribution_list") || die "cant open outfile\n";print OUT
"**faunaeuropea** $faunaeuropea_list ";close(OUT);

                        ######################################################################

                        };
                };

        }
```