

Metacoder: An R Package for Visualization and Manipulation of Community Taxonomic Diversity Data

Zachary S. L. Foster¹, Thomas J. Sharpton^{2,3,4}, Niklaus J. Grünwald^{1,4,5*}

¹ Department of Botany and Plant Pathology, Oregon State University, Corvallis, OR, 97331, USA

² Department of Microbiology, Oregon State University, Corvallis, OR, 97331, USA

³ Department of Statistics, Oregon State University, Corvallis, OR, 97331, USA

⁴ Center for Genome Research and Biocomputing, Oregon State University, Corvallis, OR, 97331, USA

⁴ Horticultural Crops Research Laboratory, USDA-ARS, Corvallis, OR, 97330, USA

* Corresponding author: nik.grunwald@ars.usda.gov

1 Abstract

2 Community-level data, the type generated by an increasing number of metabarcoding studies, is often
3 graphed as stacked bar charts or pie graphs that use color to represent taxa. These graph types do not
4 convey the hierarchical structure of taxonomic classifications and are limited by the use of color for cat-
5 egories. As an alternative, we developed *metacoder*, an R package for easily parsing, manipulating, and
6 graphing publication-ready plots of hierarchical data. *Metacoder* includes a dynamic and flexible function
7 that can parse most text-based formats that contain taxonomic classifications, taxon names, taxon identi-
8 fiers, or sequence identifiers. *Metacoder* can then subset, sample, and order this parsed data using a set of
9 intuitive functions that take into account the hierarchical nature of the data. Finally, an extremely flexible
10 plotting function enables quantitative representation of up to 4 arbitrary statistics simultaneously in a tree
11 format by mapping statistics to the color and size of tree nodes and edges. *Metacoder* also allows exploration
12 of barcode primer bias by integrating functions to run digital PCR. Although it has been designed for data
13 from metabarcoding research, *metacoder* can easily be applied to any data that has a hierarchical component
14 such as gene ontology or geographic location data. Our package complements currently available tools for
15 community analysis and is provided open source with an extensive online user manual.

Note: This article was previously submitted as a pre-print: Zachary S. L. Foster, Thomas J. Sharpton, Niklaus J. Grünwald. 2016. *Metacoder*: An R package for manipulation and heat tree visualization of community taxonomic data from metabarcoding. BioRxiv 071019; doi: <http://dx.doi.org/10.1101/071019>.

keywords: heat tree; metabarcoding; biodiversity; taxonomy; hierarchy; bioinformatics

1 Introduction

Metabarcoding is revolutionizing our understanding of complex ecosystems by circumventing the traditional limits of microbial diversity assessment, which include the need and bias of culturability, the effects of cryptic diversity, and the reliance on expert identification. Metabarcoding is a technique for determining community composition that typically involves extracting environmental DNA, amplifying a gene shared by a taxonomic group of interest using PCR, sequencing the amplicons, and comparing the sequences to reference databases [1]. It has been used extensively to explore communities inhabiting diverse environments, including oceans [2], plants [3], animals [4], humans [5], and soil [6].

The complex community data produced by metabarcoding is challenging conventional graphing techniques. Most often, bar charts, stacked bar charts, or pie graphs are employed that use color to represent a small number of taxa at the same rank (e.g. phylum, class, etc). This reliance on color for categorical information limits the number of taxa that can be effectively displayed, so most published figures only show results at a coarse taxonomic rank (e.g. class) or for only the most abundant taxa. These graphing techniques do not convey the hierarchical nature of taxonomic classifications, potentially obscuring patterns in unexplored taxonomic ranks that might be more biologically important. More recently, tree-based visualizations are becoming available as exemplified by the python-based MetaPhlAn and the corresponding graphing software GraPhlAn [7]. This tool allows visualization of high-quality circular representations of taxonomic trees.

Here, we introduce the R package *metacoder* that is specifically designed to address some of these problems in metabarcoding-based community ecology, focusing on parsing and manipulation of hierarchical data and community visualization in R. *Metacoder* provides a visualization that we call “heat trees” which quantitatively depicts statistics associated with taxa, such as abundance, using the color and size of nodes and edges in a taxonomic tree. These heat trees are useful for evaluating taxonomic coverage, barcode bias, or displaying differences in taxon abundance between communities. To import and manipulate data, *metacoder* provides

41 a means of extracting and parsing taxonomic information from text-based formats (e.g. reference database
42 FASTA headers) and an intuitive set of functions for subsetting, sampling, and rearranging taxonomic data.
43 *Metacoder* also allows exploration of barcode primer bias by integrating digital PCR. All this functionality
44 is made intuitive and user-friendly while still allowing extensive customization and flexibility. *Metacoder*
45 can be applied to any data that can be organized hierarchically such as gene ontology or geographic loca-
46 tion. *Metacoder* is an open source project available on CRAN and is provided with comprehensive online
47 documentation including examples.

48 2 Design and Implementation

49 The R package *metacoder* provides a set of novel tools designed to parse, manipulate, and visualize community
50 diversity data in a tree format using any taxonomic classification (Figure 1). Figure 1 illustrates the ease of
51 use and flexibility of *metacoder*. It shows an example analysis extracting taxonomy from the 16S Ribosomal
52 Database Project (RDP) training set for *mothur* [8], filtering and sampling the data by both taxon and
53 sequence characteristics, running digital PCR, and graphing the proportion of sequences amplified for each
54 taxon. Table 1 provides an overview of the core functions available in *metacoder*.

55 **Fig. 1. *Metacoder* has an intuitive and easy to use syntax.** The code in this example analysis parses
56 the taxonomic data associated with sequences from the Ribosomal Database Project [9] 16S training set,
57 filters and subsamples the data by sequence and taxon characteristics, conducts digital PCR, and displays
58 the results as a heat tree. All functions in bold are from the *metacoder* package. Note how columns and
59 functions in the `taxmap` object (green box) can be referenced within functions as if they were independent
60 variables.

61 2.1 The `taxmap` data object

62 To store the taxonomic hierarchy and associated observations (e.g. sequences) we developed a new data object
63 class called `taxmap`. The `taxmap` class is designed to be as flexible and easily manipulated as possible. The
64 only assumption made about the users data is that it can be represented as a set of observations assigned
65 to a hierarchy; the hierarchy and the observations do not need to be biological. The class contains two
66 tables in which user data is stored: a taxonomic hierarchy stored as an edge list of unique IDs and a set
67 of observations mapped to that hierarchy (Figure 1). Users can add, remove, or reorder both columns and
68 rows in either `taxmap` table using convenient functions included in the package (Table 1). For each table,

69 there is also a list of functions stored with the class that each create a temporary column with the same
70 name when referenced by one of the manipulation or plotting functions. These are useful for attributes that
71 must be updated when the data is subset or otherwise modified, such as the number of observations for each
72 taxon (see “n_obs” in Figure 1). If this kind of derived information was stored in a static column, the user
73 would have to update the column each time the data set is subset, potentially leading to mistakes if this is
74 not done. There are many of these column-generating functions included by default, but the user can easily
75 add their own by adding a function that takes a `taxmap` object. The names of columns or column-generating
76 functions in either table of a `taxmap` object can be referenced as if they were independent variables in most
77 *metacoder* functions in the style of popular R packages like *ggplot2* and *dplyr*. This makes the code much
78 easier to read and write.

79 **2.2 Universal parsing and retrieval of taxonomic information**

80 *Metacoder* provides a way to extract taxonomic information from text-based formats so it can be manipu-
81 lated within R. One of the most inefficient steps in bioinformatics can be loading and parsing data into a
82 standardized form that is usable for computational analysis. Many databases have unique taxonomy formats
83 with differing types of taxonomic information. The structure and nomenclature of the taxonomy used can
84 be unique to the database or reference another database such as GenBank [10]. Rather than creating a
85 parser for each data format, *metacoder* provides a single function to parse any format definable by regular
86 expressions that contains taxonomic information (Figure 1). This makes it easier to use multiple data sources
87 with the same downstream analysis.

88 The `extract_taxonomy` function can parse hierarchical classifications or retrieve classifications from online
89 databases using taxon names, taxon IDs, or Genbank sequence IDs. The user supplies a regular expression
90 with capture groups (parentheses) and a corresponding key to define what parts of the input can provide
91 classification information. The `extract_taxonomy` function has been used successfully to parse several major
92 database formats including Genbank [10], UNITE [11], Protist Ribosomal Reference Database (PR2) [12],
93 Greengenes [13], Silva [14], and, as illustrated in figure 1, the RDP [9]. Examples for each database are
94 provided in the user manuals [15].

Table 1: Primary functions found in *metacoder*.

| Function | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • <code>extract_taxonomy</code> | Parses taxonomic data from arbitrary text and returns a <code>taxmap</code> object containing a table with rows corresponding to inputs (i.e. observations) and a table with rows corresponding to taxa. |
| <ul style="list-style-type: none"> • <code>heat_tree</code> | Makes tree-based plots of data stored in <code>taxmap</code> objects. Color, size, and labels of tree components can be mapped to arbitrary data. The output is a <i>ggplot2</i> object. |
| <ul style="list-style-type: none"> • <code>primersearch</code> | Executes the EMBOSS program <i>primersearch</i> on sequence data stored in a <code>taxmap</code> object. Results are parsed, added to the input <code>taxmap</code> object and returned. |
| <ul style="list-style-type: none"> • <code>mutate_taxa</code> • <code>mutate_obs</code> • <code>transmute_taxa</code> • <code>transmute_obs</code> | Modify or add columns of taxon or observation data in <code>taxmap</code> objects. <code>mutate.*</code> adds columns and <code>transmute.*</code> returns only new columns. |
| <ul style="list-style-type: none"> • <code>select_taxa</code> • <code>select_obs</code> | Subset columns of taxon or observation data in <code>taxmap</code> objects. |
| <ul style="list-style-type: none"> • <code>filter_taxa</code> • <code>filter_obs</code> | Subset rows of taxon or observation data in <code>taxmap</code> objects based on arbitrary conditions. Hierarchical relationships among taxa and mappings between taxa and observations are taken into account. |
| <ul style="list-style-type: none"> • <code>arrange_taxa</code> • <code>arrange_obs</code> | Order rows of taxon or observation data in <code>taxmap</code> objects. |
| <ul style="list-style-type: none"> • <code>sample_n_taxa</code> • <code>sample_n_obs</code> • <code>sample_frac_taxa</code> • <code>sample_frac_obs</code> | Randomly subsample rows of taxon or observation data in <code>taxmap</code> objects. Weights can be applied that take into account the taxonomic hierarchy and associated observations. Hierarchical relationships among taxa and mappings between taxa and observations are taken into account. |
| <ul style="list-style-type: none"> • <code>subtaxa</code> • <code>supertaxa</code> • <code>observations</code> • <code>roots</code> | Returns the indices of rows in taxon or observation data in <code>taxmap</code> objects. Used to map taxa to related taxa and observations. |

95 2.3 Intuitive manipulation of taxonomic data

96 *Metacoder* makes it easy to subset and sample large data sets composed of thousands of observations (e.g. se-
97 quences) assigned to thousands of taxa, while taking into account hierarchical relationships. This allows for
98 exploration and analysis of manageable subsets of a large data set. Taxonomies are inherently hierarchical,
99 making them difficult to subset and sample intuitively compared with typical tabular data. In addition
100 to the taxonomy itself, there is usually also data assigned to taxa in the taxonomy, which we refer to as
101 “observations”. Subsetting either the taxonomy or the associated observations, depending on the goal, might
102 require subsetting both to keep them in sync. For example, if a set of taxa are removed or left out of a
103 random subsample, should the subtaxa and associated observations also be removed, left as is, or reassigned
104 to a supertaxon? If observations are removed, should the taxa they were assigned to also be removed? The
105 functions provided by *metacoder* gives the user control over these details and simplifies their implementation.

106 *Metacoder* allows users to intuitively and efficiently subset complex hierarchical data sets using a cohesive
107 set of functions inspired by the popular *dplyr* data-manipulation philosophy. *Dplyr* is an R package for
108 providing a conceptually consistent set of operations for manipulating tabular information [16]. Whereas
109 *dplyr* functions each act on a single table, *metacoder*’s analogous functions act on both the taxon and
110 observation tables in a `taxmap` object (Table 1). For each major *dplyr* function there are two analogous
111 *metacoder* functions: one that manipulates the taxon table and one that manipulates the observations table.
112 The functions take into account the relationship between the two tables and can modify both depending
113 on parameterization, allowing for operations on taxa to affect their corresponding observations and vice
114 versa. They also take into account the hierarchical nature of the taxon table. For example, the *metacoder*
115 functions `filter_taxa` and `filter_obs` are based on the *dplyr* function `filter` and are used to remove rows
116 in the taxon and observation tables corresponding to some criterion. Unlike simply applying a filter to these
117 tables directly, these functions allow the subtaxa, supertaxa, and/or observations of taxa passing the filter
118 to be preserved or discarded, making it easy to subset the data in diverse ways (Figure 1). There are also
119 functions for ordering rows (`arrange_taxa`, `arrange_obs`), subsetting columns (`select_taxa`, `select_obs`),
120 and adding columns (`mutate_taxa`, `mutate_obs`).

121 *Metacoder* also provides functions for random sampling of taxa and corresponding observations. The function
122 `taxonomic_sample` is used to randomly sub-sample items such that all taxa of one or more given ranks have
123 some specified number of observations representing them. Taxa with too few sequences are excluded and
124 taxa with too many are randomly subsampled. Whole taxa can also be sampled based on the number of

125 sub-taxa they have. Alternatively, there are *dplyr* analogues called `sample_n_taxa` and `sample_n_obs`, which
126 can sample some number of taxa or observations. In both functions, weights can be assigned to taxa or
127 observations, influencing how likely each is to be sampled. For example, the probability of sampling a given
128 observation can be determined by a taxon characteristic, such as the number of observations assigned to
129 that taxon, or it could be determined by an observation characteristic, like sequence length. Similar to
130 the `filter_*` functions, there are parameters controlling whether selected taxa's subtaxa, supertaxa, or
131 observations are included or not in the sample (Figure 1).

132 2.4 Heat tree plotting of taxonomic data

133 Visualizing the massive data sets being generated by modern sequencing of complex ecosystems is typically
134 done using traditional stacked barcharts or pie graphs, but these ignore the hierarchical nature of taxonomic
135 classifications and their reliance on colors for categories limits the number of taxa that can be distinguished
136 (Figure 2). Generic trees can convey a taxonomic hierarchy, but displaying how statistics are distributed
137 throughout the tree, including internal taxa, is difficult. *Metacoder* provides a function that plots up to
138 4 statistics on a tree with quantitative legends by automatically mapping any set of numbers to the color
139 and width of nodes and edges. The size and content of edge and node labels can also be mapped to custom
140 values. These publication-quality graphs provide a method for visualizing community data that is richer than
141 is currently possible with stacked bar charts. Although there are other R packages that can plot variables on
142 trees, like *phyloseq* [17], these have been designed for phylogenetic rather than taxonomic trees and therefore
143 optimized for plotting information on the tips of the tree and not on internal nodes.

144 **Fig. 2. Heat trees allow for a better understanding of community structure than stacked bar**
145 **charts.** The stacked bar chart on the left represents the abundance of organisms in two samples from the
146 Human Microbiome Project [5]. The same data are displayed as heat trees on the right. In the heat trees,
147 size and color of nodes and edges are correlated with the abundance of organisms in each community. Both
148 visualizations show communities dominated by firmicutes, but the heat trees reveal that the two samples
149 share no families within firmicutes and are thus much more different than suggested by the stacked bar chart.
150 The function `heat_tree` creates a tree utilizing color and size to display taxon statistics (e.g., sequence
151 abundance) for many taxa and ranks in one intuitive graph (Figure 2). Taxa are represented as nodes and
152 both color and size are used to represent any statistic associated with taxa, such as abundance. Although the
153 `heat_tree` function has many options to customize the appearance of the graph, it is designed to minimize

154 the amount of user-defined parameters necessary to create an effective visualization. The size range of
155 graph elements is optimized for each graph to minimize overlap and maximize size range. Raw statistics are
156 automatically translated to size and color and a legend is added to display the relationship. Unlike most
157 other plotting functions in R, the plot looks the same regardless of output size, allowing the graph to be
158 saved at any size or used in complex, composite figures without changing parameters. These characteristics
159 allow `heat_tree` to be used effectively in pipelines and with minimal parameterization since a small set of
160 parameters displays diverse taxonomy data. The output of the `heat_tree` function is a *ggplot2* object, making
161 it compatible with many existing R tools. Another novel feature of heat trees is the automatic plotting of
162 multiple trees when there are multiple “roots” to the hierarchy. This can happen when, for example, there
163 are “Bacteria” and “Eukaryota” taxa without a unifying “Life” taxon, or when coarse taxonomic ranks are
164 removed to aid in the visualization of large data sets (Figure 3).

165 **Fig. 3. Heat trees display up to four metrics in a taxonomic context and can plot multiple**
166 **trees per graph.** Most graph components, such as the size and color of text, nodes, and edges, can be
167 automatically mapped to arbitrary numbers, allowing for a quantitative representation of multiple statistics
168 simultaneously. This graph depicts the uncertainty of OTU classifications from the TARA global oceans
169 survey [2]. Each node represents a taxon used to classify OTUs and the edges determine where it fits in
170 the overall taxonomic hierarchy. Node diameter is proportional to the number of OTUs classified as that
171 taxon and edge width is proportional to the number of reads. Color represents the percent of OTUs assigned
172 to each taxon that are somewhat similar to their closest reference sequence (>90% sequence identity). **a.**
173 Metazoan diversity in detail. **b.** All taxonomic diversity found. Note that multiple trees are automatically
174 created and arranged when there are multiple roots to the taxonomy.

175 3 Results

176 3.1 Heat trees allow quantitative visualization of community diversity data

177 We developed heat trees to allow visualization of community data in a taxonomic context by mapping any
178 statistic to the color or size of tree components. Here, we reanalyzed data set 5 from the TARA oceans
179 eukaryotic plankton diversity study to visualize the similarity between OTUs observed in the data set and
180 their closest match to a sequence in a reference database [2]. The TARA ocean expedition analyzed DNA
181 extracted from ocean water throughout the world. Even though a custom reference database was made using

182 curated 18S sequences spanning all known eukaryotic diversity, many of the OTUs observed had no close
183 match. Figure 3 shows a heat tree that illustrates the proportion of OTUs that were well characterized in
184 each taxon (at least 90% identical to a reference sequence). Color indicates the percentage of OTUs that
185 are well characterized, node width indicates the number of OTUs assigned to each taxon, and edge width
186 indicates the number of reads. Taxa with ambiguous names and those with less than 200 reads have been
187 filtered out for clarity. This figure illustrates one of the principal advantages of heat trees, as it reveals many
188 clades in the tree that contain only red lineages, which indicate that the entire taxonomic group is poorly
189 represented in the reference sequence database. Of particular interest are those clades with predominantly
190 red lineages that also have relatively large nodes, such as Harpacticoida (in Copepoda on the right). These
191 represent taxonomic groups that were found to have high amounts of diversity in the oceans, but for which
192 we have a paucity of genomic information. Investigators interested in improving the genomic resolution of
193 the biosphere can thus use these approaches to rapidly assess which taxa should be prioritized for focused
194 investigations. Note that a large portion of the taxa shown in red, yellow or orange have many OTUs with
195 a poor match to the reference taxonomic hierarchy.

196 **3.2 Flexible parsing allows for similar use of diverse data**

197 Metabarcoding studies often rely on techniques or data that may introduce bias into an investigation. For
198 example, the specific set of PCR primers used to amplify genomic DNA and the taxonomic annotation
199 database can both have an effect on the study results. A quick and inexpensive way to estimate biases
200 caused by primers is to use digital PCR, which simulates PCR success using alignments between reference
201 sequences and primers. *Metacoder* can be used to explore different databases or primer combinations to
202 assess these effects since it supplies functions to parse diverse data sources, conduct digital PCR, and plot
203 the results. Figure 4 shows a series of heat tree comparisons that were produced using a common 16S
204 rRNA metabarcoding primer set [18] and digital PCR against the full-length 16S sequences found in three
205 taxonomic annotation databases: Greengenes [13], RDP [9], and SILVA [14]. These heat trees reveal subsets
206 of the full taxonomies for these three databases that poorly amplify by digital PCR using the selected
207 primers. As a result, they indicate which lineages within each of the taxonomies may be challenging to
208 detect in a metabarcoding study that uses these primers. Importantly, different sets of primers likely amplify
209 different sets of taxa, so investigators interested in specific lineages can use this approach in conjunction with
210 various primer sets to identify those that maximize the likelihood of discovery and reduce wasted sequencing
211 resources on non-target organisms. However, these heat maps do not indicate whether one database is

212 necessarily preferable over another, as they differ in the structure of their taxonomies, as well as the number
213 and phylogenetic diversity of their reference sequences. For example, most of the bacterial clades that do
214 not amplify well in the SILVA lineages are unnamed lineages that are not found in the other databases,
215 indicating that they warrant further exploration.

216 **Fig. 4. Flexible parsing and digital PCR allows for comparisons of primers and databases.**
217 Shown is a comparison of digital PCR results for three 16S reference databases. The plots on the left display
218 abundance of all bacterial 16S sequences. Plots on the right display all taxa with subtaxa not entirely
219 amplified by digital PCR using universal 16S primers [19]. Node color and size display the proportion and
220 number of sequences not amplified respectively.

221 **3.3 Heat trees can show pairwise comparisons of communities across treatments**

222 One challenge in metabarcoding studies is visually determining how specific sub-sets of samples vary in
223 their taxonomic composition. Unlike most other graphing software in R, *metacoder* produces graphs that
224 look the same at any output size or aspect ratio, allowing heat trees to be easily integrated into larger
225 composite figures without changing the code for individual subplots. Using color to depict the difference in
226 read or OTU abundance between two treatments can result in particularly effective visualizations, especially
227 when the presence of color is made dependent on a statistical test. To examine more than two treatments
228 at once, a matrix of these kind of heat trees can be combined with a labeled “guide” tree. Figure 5 shows
229 application of this idea to human microbiome data showing pairwise differences between body sites. Coloring
230 indicates significant differences between the median proportion of reads for samples from different body sites
231 as determined using a Wilcox rank-sum test followed by a Benjamini-Hochberg (FDR) correction for multiple
232 testing. The intensity of the color is relative to the log-2 ratio of difference in median proportions. Brown
233 taxa indicate an enrichment in body sites listed on the top of the graph and green is the opposite. While the
234 original study [5] showed abundance plots, our visualization provides the taxonomic context. For example,
235 *Haemophilus*, *Streptococcus*, and *Prevotella* spp. are enriched in saliva (brown) relative to stool where
236 *Bacteroides* is enriched (green). We also see that in the Lachnospiraceae clade several genera shown in both
237 green and brown taxa are differentially abundant. These observations are consistent with known differences
238 in the human-associated microbiome across body sites, but heat trees uniquely provide an integrated view
239 of how all levels of a taxonomy vary for all pairs of body sites.

240 **Fig. 5. Scale-independent appearance facilitates complex, composite figures.** All graph compo-

241 nents, including text, have the same relative sizes independent of output size, unlike most graphical packages
242 in R, making it easier to create composite figures entirely within R. This graph uses 16S metabarcoding data
243 from the human microbiome project study. The gray tree on the lower left functions as a key for the smaller
244 unlabeled trees. The color of each taxon represents the log-2 ratio of median proportions of reads observed at
245 each body site. Only significant differences are colored, determined using a Wilcox rank-sum test followed by
246 a Benjamini-Hochberg (FDR) correction for multiple comparisons. For example, *Haemophilus*, *Streptococcus*,
247 *Prevotella* are enriched in saliva (brown) relative to stool where *Bacteroides* is enriched (green).

248 3.4 Other applications

249 The `taxmap` data object defined in *metacoder* can be used for any data that can be classified by a hierarchy.
250 Figure 6, for example, shows an analysis of votes cast in the 2016 US Democratic party national primaries
251 organized by geography. The heat tree reveals distinct patterns such as a sweep by Clinton in the South and
252 a split on the West coast, with California predominantly voting for Clinton while Washington and Oregon
253 predominantly voted for Sanders. Another potential application is displaying the results of gene expression
254 studies by associating differential expression with gene ontology (GO) annotations. Figure 7 shows the
255 results of a RNA-seq study on the effect of glucocorticoids on smooth muscle tissue [20]. All biological
256 processes influenced by at least one gene with a significant change in expression are plotted. The authors
257 of the study find that genes involved in immune response are influenced by the glucocorticoid treatment.
258 Viewing these results in a heat tree shows not only the specific immune process affected (the branch on the
259 middle right), but also the more general phenomena they constitute; regulation of high level phenomena,
260 like immune system function, can be explained by specific processes like “leukocyte chemotaxis” and these
261 specific processes are put into the context of the phenomena they contribute to. This is more informative
262 than simply reporting the results for a single level of the GO annotation hierarchy or discussing the effects
263 of genes one at a time.

264 **Fig. 6. *Metacoder* can be used with any type of data that can be organized hierarchi-**
265 **cally.** This plot shows the results of the 2016 Democratic primary election organized by region, divi-
266 sion, state, and county. The regions and divisions are those defined by the United States census bureau.
267 Color corresponds to the difference in the percentage of votes for candidates Hillary Clinton (green) and
268 Bernie Sanders (brown). Size corresponds to the total number of votes cast. Data was downloaded from
269 <https://www.kaggle.com/benhamner/2016-us-election/>.

270 **Fig. 7. Another alternate use example: vizualizing gene expression data in a GO hierarchy.**
271 The gene ontology for all differentially expressed genes in a study on the effect of a glucocorticoid on airway
272 smooth muscle tissue [20]. Color indicates the sign and intensity of averaged changes in gene expression and
273 the size indicates the number of genes classified by a given gene ontology term.

274 4 Availability and Future Directions

275 The R package *metacoder* is an open-source project under the MIT License. Stable releases of *metacoder* are
276 available on CRAN while recent improvements can be downloaded from github (<https://github.com/grunwaldlab/metacoder>).
277 A manual with documentation and examples is provided [15]. This manual also provides the code to repro-
278 duce all figures included in this manuscript.

279 We are currently continuing development of *metacoder*. We welcome contributions and feedback from the
280 community. We want to make *metacoder* functions and classes compatible with those from other bioinforma-
281 tic R packages such as *phyloseq*, *ape*, *seqinr*, and *taxize*. We might integrate more options for digital PCR
282 and barcode gap analysis, perhaps using *ecoPCR* or the R packages *PrimerMiner* and *Spider*. We are also
283 considering adding additional visualization functions.

284 5 Acknowledgments

285 This work was supported in part by funds from USDA ARS CRIS Project 2027-22000-039-00 and the USDA
286 ARS Floriculture Nursery Research Initiative. The use of trade, firm, or corporation names in this publication
287 is for the information and convenience of the reader. Such use does not constitute an official endorsement
288 or approval by the United States Department of Agriculture or the Agricultural Research Service of any
289 product or service to the exclusion of others that may be suitable.

290 6 Author Contributions

291 Conceived and designed the experiments: ZSLF, NJG, TJS. Performed the experiments: ZSLF. Analyzed
292 the data: ZSLF. Contributed reagents/materials/analysis tools: ZSLF, NJG. Wrote the paper: ZSLF, NJG,
293 TJS. Designed, developed scripts: ZSLF.

294 References

- 295 1. Cristescu ME. From barcoding single individuals to metabarcoding biological communities: towards an
296 integrative approach to the study of global biodiversity. *Trends Ecol Evol.* 2014;29: 566–571.
- 297 2. De Vargas C, Audic S, Henry N, Decelle J, Mahé F, Logares R, et al. Eukaryotic plankton diversity in
298 the sunlit ocean. *Science.* 2015;348: 1261605.
- 299 3. Coleman-Derr D, Desgarenes D, Fonseca-Garcia C, Gross S, Clingenpeel S, Woyke T, et al. Plant
300 compartment and biogeography affect microbiome composition in cultivated and native *Agave* species. *New*
301 *Phytol.* 2016;209: 798–811.
- 302 4. Yu DW, Ji Y, Emerson BC, Wang X, Ye C, Yang C, et al. Biodiversity soup: metabarcoding of arthropods
303 for rapid biodiversity assessment and biomonitoring. *Methods Ecol Evol.* 2012;3: 613–623.
- 304 5. Consortium HMP, others. Structure, function and diversity of the healthy human microbiome. *Nature.*
305 2012;486: 207–214.
- 306 6. Gilbert JA, Jansson JK, Knight R. The Earth Microbiome project: successes and aspirations. *BMC Biol.*
307 2014;12: 1.
- 308 7. Segata N, Waldron L, Ballarini A, Narasimhan V, Jousson O, Huttenhower C. Metagenomic microbial
309 community profiling using unique clade-specific marker genes. *Nature Methods.* 2012;9: 811–814.
- 310 8. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, et al. Introducing mothur:
311 open-source, platform-independent, community-supported software for describing and comparing microbial
312 communities. *Appl Environ Microbiol.* 2009;75: 7537–7541.
- 313 9. Maidak BL, Olsen GJ, Larsen N, Overbeek R, McCaughey MJ, Woese CR. The ribosomal database
314 project (RDP). *Nucleic Acids Res.* 1996;24: 82–85.
- 315 10. Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, et al. GenBank. *Nucleic*
316 *Acids Res.* 2013;41: D36–D42.
- 317 11. Kõljalg U, Nilsson RH, Abarenkov K, Tedersoo L, Taylor AFS, Bahram M, et al. Towards a unified
318 paradigm for sequence-based identification of fungi. *Mol Ecol.* 2013;22: 5271–5277.
- 319 12. Guillou L, Bachar D, Audic S, Bass D, Berney C, Bittner L, et al. The Protist Ribosomal Reference
320 database (PR2): a catalog of unicellular eukaryote small sub-unit rRNA sequences with curated taxonomy.

321 Nucleic Acids Res. 2012;41: D597–D604.

322 13. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K, et al. Greengenes, a chimera-
323 checked 16S rRNA gene database and workbench compatible with ARB. Appl Environ Microbiol. 2006;72:
324 5069–5072.

325 14. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, et al. The SILVA ribosomal RNA gene
326 database project: improved data processing and web-based tools. Nucleic Acids Res. 2013;41: D590–D596.

327 15. Foster ZSL, Grünwald NJ. Metacoder user documentation [Internet]. 2016. doi:10.5281/zenodo.158228

328 16. Wickham H, Francois R. dplyr: A Grammar of Data Manipulation [Internet]. 2016. Retrieved:
329 <https://CRAN.R-project.org/package=dplyr>

330 17. McMurdie PJ, Holmes S. phyloseq: an R package for reproducible interactive analysis and graphics of
331 microbiome census data. PloS One. 2013;8: e61217.

332 18. Caporaso JG, Lauber CL, Walters WA, Berg-Lyons D, Huntley J, Fierer N, et al. Ultra-high-throughput
333 microbial community analysis on the Illumina HiSeq and MiSeq platforms. ISME J. 2012;6: 1621–1624.

334 19. Walters W, Hyde ER, Berg-Lyons D, Ackermann G, Humphrey G, Parada A, et al. Improved Bacterial
335 16S rRNA Gene (V4 and V4-5) and Fungal Internal Transcribed Spacer Marker Gene Primers for Microbial
336 Community Surveys. mSystems. 2016;1: e00009–15.

337 20. Himes BE, Jiang X, Wagner P, Hu R, Wang Q, Klanderman B, et al. RNA-Seq transcriptome profiling
338 identifies CRISPLD2 as a glucocorticoid responsive gene that modulates cytokine function in airway smooth
339 muscle cells. PloS One. 2014;9: e99625.

Functions

extract_taxonomy

```
# Load package
library(metacoder)

# Parse taxonomic data from sequence headers
data <- extract_taxonomy(seqs, regex = "^(.*)\\t(.*)",
                        key = c(id = "obs_info", "class"),
                        class_sep = ";")
```

Manipulation / Analysis

```
# Filter, subsample, and run in silico PCR
pcr <- filter_taxa(data, n_obs > 1) %>%
  filter_obs(nchar(sequence) > 1000) %>%
  filter_taxa(name == "Bacteria", subtaxa = TRUE) %>%
  sample_n_obs(5000, taxon_weight = 1 / n_obs) %>%
  sample_n_taxa(1000, supertaxa = TRUE) %>%
  primersearch(forward = "CTCCTACGGGAGGCAGCAG",
              reverse = "GAATTACCGCGGCKGCTG",
              mismatch = 10) %>%
  filter_taxa(prop_amplified < 0.9, supertaxa = TRUE)
```

heat_tree

```
# Plot results of in silico PCR
heat_tree(pcr, node_size = n_obs, node_label = name,
         node_color = prop_amplified,
         node_color_range = c("red", "yellow", "cyan"))
```

Data Structures

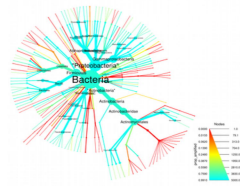
Raw input

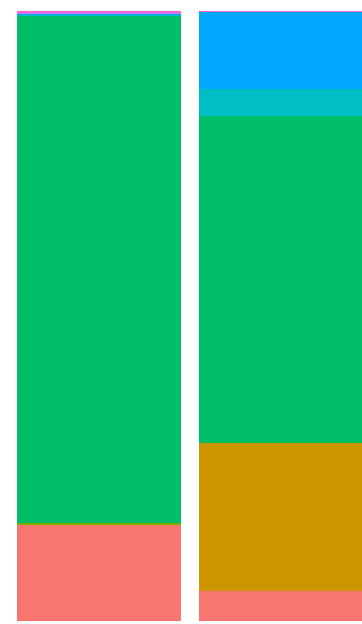
```
# Load FASTA file of RDP 16S training set
seqs <- ape::read.FASTA("rdp_training_set.fa")
print(names(seqs)[1])
AB294171_S001198039 ROOT;BACTERIA;FIRMICUTES;BACILLI;LACTOBACI...
```

taxmap

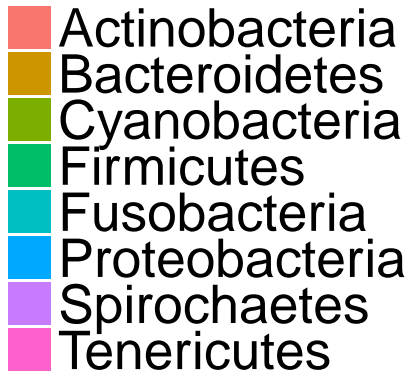
```
print(data)
`TAXMAP` WITH 2794 TAXA AND 10650 OBSERVATIONS:
----- TAXA -----
1, 2, 3, 4 ... 2791, 2792, 2793, 2794
----- TAXON_DATA -----
  TAXON_IDS SUPERTAXON_IDS NAME
    <CHR>      <CHR>      <CHR>
1         1             <NA>  ROOT
2         2             1  ARCHAEA
3         3             1  BACTERIA
# ... WITH 2,791 MORE ROWS
----- OBS_DATA -----
  OBS_TAXON_IDS ID
    <CHR>      <CHR>
1        1289 AB294171_S001198039
2         369 AB243007_S000622964
3        2402 AJ717394_S000623308
# ... WITH 1.065E+04 MORE ROWS, AND 1 MORE VARIABLES:
# SEQUENCE <CHR>
----- TAXON_FUNCS -----
N_OBS ... N_SUBTAXA_1, HIERARCHIES
```

ggplot2 object

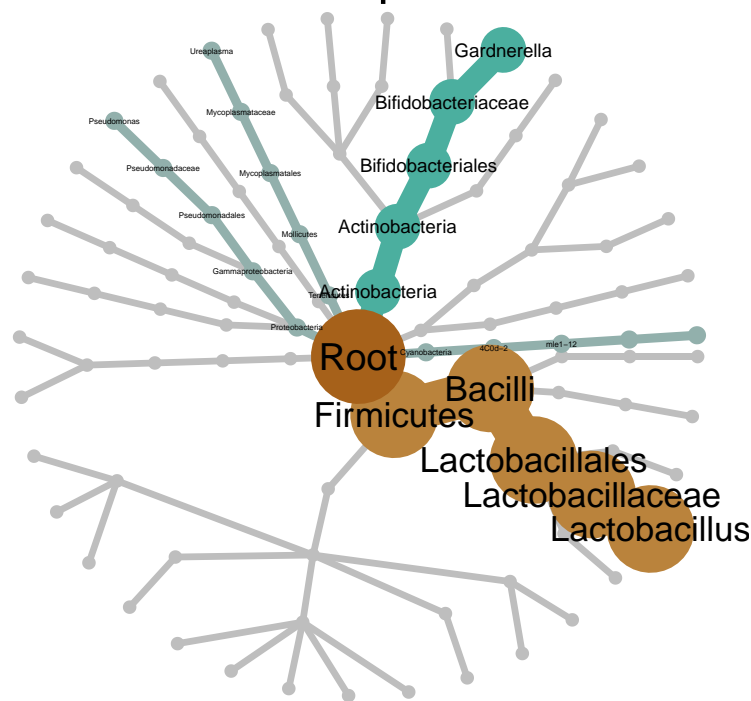




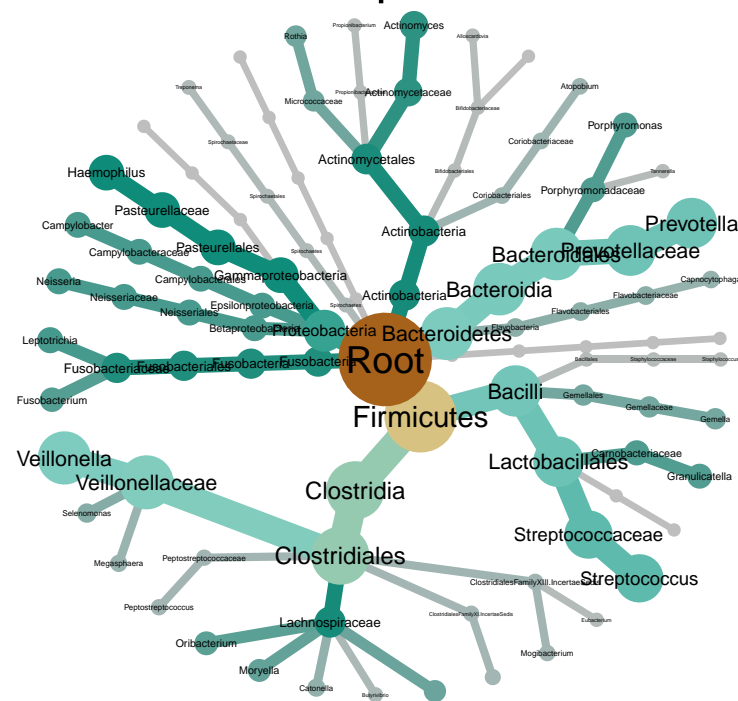
1 2
Sample

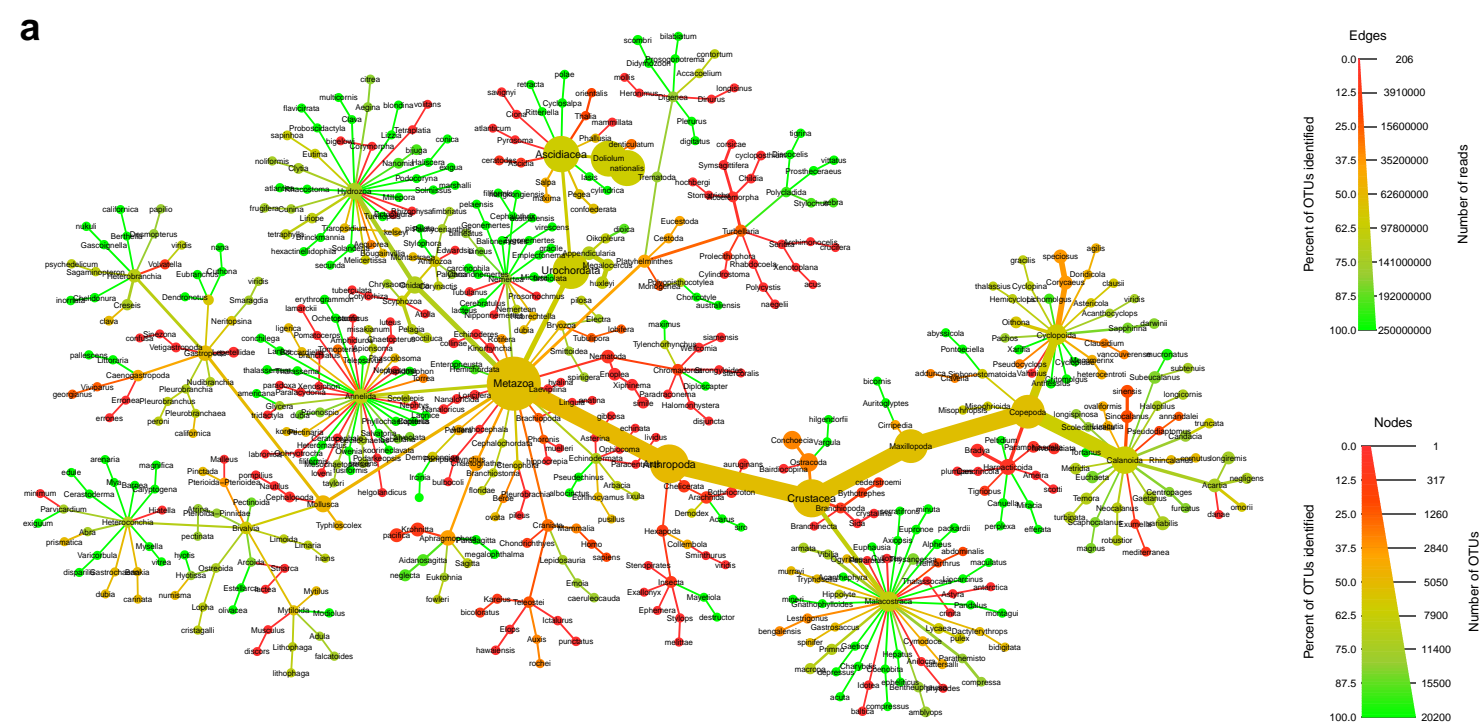
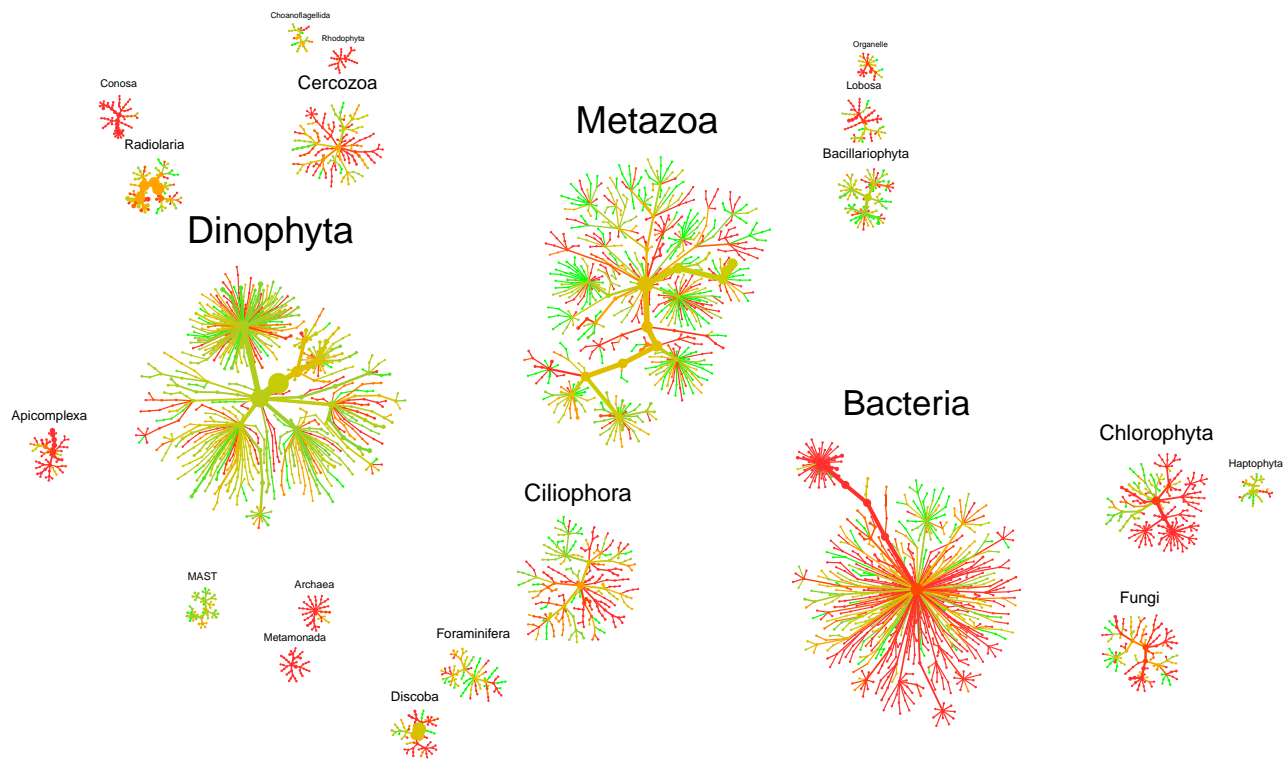


Sample 1



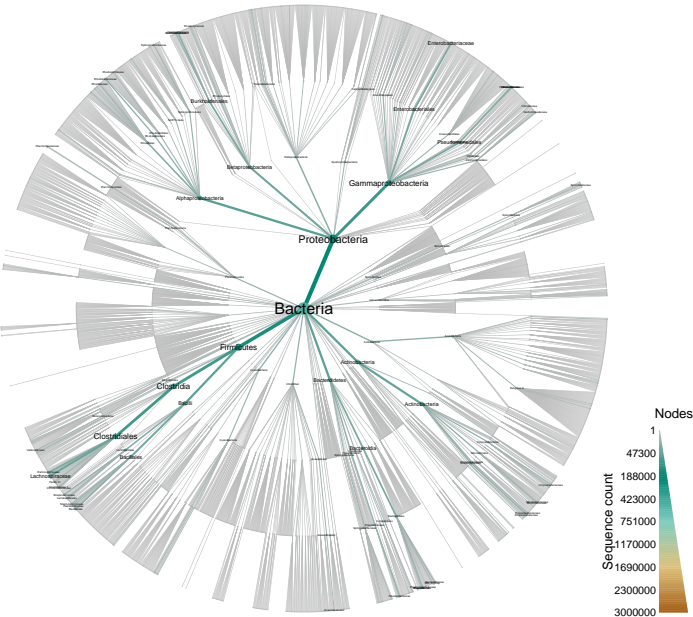
Sample 2



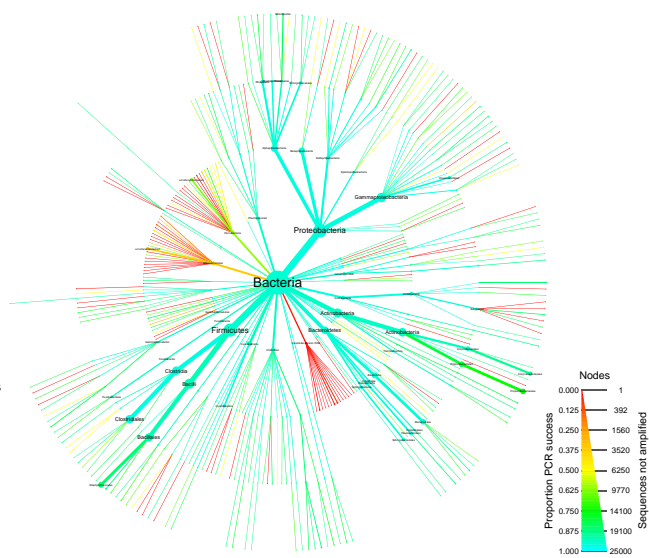
a**b**

All data

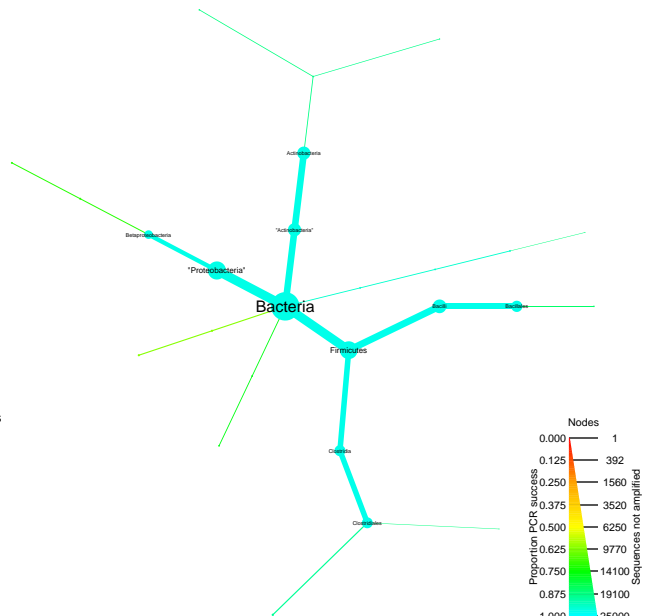
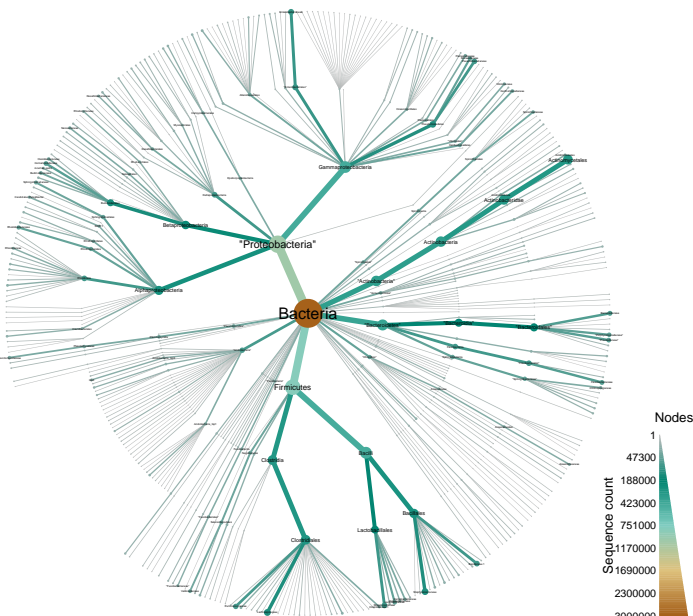
SILVA



Not amplified



RDP



Greengenes

