PHYLOmap: A program for drawing heatmap with phylogenetic tree for abundance tables

Umer Zeeshan Ijaz and Christopher Quince
School of Engineering, University of Glasgow, UK
Umer.Ijaz@glasgow.ac.uk
http://userweb.eng.gla.ac.uk/umer.ijaz

March 30, 2014

Abstract

In metagenomic studies, there are two popular methods to characterise diversity and richness of microbial populations in environmental samples: sequencing data from small-subunit ribosomal RNA (SSU rRNA); and large-scale shotgun whole-genome sequencing. The taxonomic analysis software for these methods use the lowest common ancestor (LCA) based taxonomic classification method by searching sequences against public databases, assigning them to the most similar taxa in a given taxonomy, and generating the abundance tables. These tables in post-hoc analysis are visualised in different ways to discriminate between environmental samples, most frequently as heatmaps and ordination diagrams. We developed PHYLOmap, which utilises the interactive Tree of Life (iTOL) API and displays heatmaps with phylogenetic trees and offers better visualisation to conventional heatmaps. In this document, we discuss strategies to integrate PHYLOmap with the data generated from frequently used taxonomic classification software in metagenomic analysis.

1 Introduction

iTOL [4] is an online tool for the display and manipulation of phylogenetic trees in the Newick format. It offers standard and circular tree representation, and several functions on the website that allow datasets to be uploaded (using `http://itol.embl.de/batch_uploader.cgi` with a standard `POST` request) with their corresponding trees, and the ability to customise the tree displays in different ways. Each tree display can then be exported (using `http://itol.embl.de/batch_downloader.cgi` with `POST` or `GET` request) to several graphical formats, both bitmap and vector based. Moreover, by parameterisation of the web request, the exported tree displays can be modified to exclude user-selected leaf nodes and to collapse internal nodes. An important feature of the website is a tree generator based on NCBI taxonomy (`http://itol.embl.de/other_trees.shtml`) through which generation of complete sub-trees for user-provided list of NCBI's taxonomy IDs or scientific names is possible.
MEGAN [5] is a *de facto* analysis tool for metagenomic of short-read shotgun sequencing data. With version 2 onwards, there is support to compare a collection of different datasets visually through a "comparison view" based on a tree in which each node shows the number of reads assigned to it for each of the datasets as a heatmap. However, to construct such a view, the datasets must first be individually opened in the program. For a few datasets, this interactivity is acceptable, however, it is also vital, that some form of automation is also ensured, attenuating the growing complexity for analysis of thousands of datasets. Eventually, the presence or absence of interactivity depends on the goals of the user and the overall conditions in which such interaction is allowed. For example, in the Earth Microbiome Project [3], which aims to catalog microbial communities across the globe based on thousands of metagenomic samples, an automated way of generating visualisations becomes necessary. Through PHYLOmap, we endeavour to achieve this automation by providing a Linux shell based workflow that can be incorporated in standard bioinformatic pipelines as well as a generalized approach to support many other taxonomies such as `SilvaMod` and `Greengenes`.

1.1 Availability

The software and accompanying utilities are freely available under a GNU General Public License (v3) from
PHYLOmap.py: `http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/PHYLOmap.py`
collateResults.pl: `http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/collateResults.pl`
Path2Newick.java: `http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/Path2Newick.java`
Example datasets used in this document: `http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/PHYLOmap_example_datasets.zip`

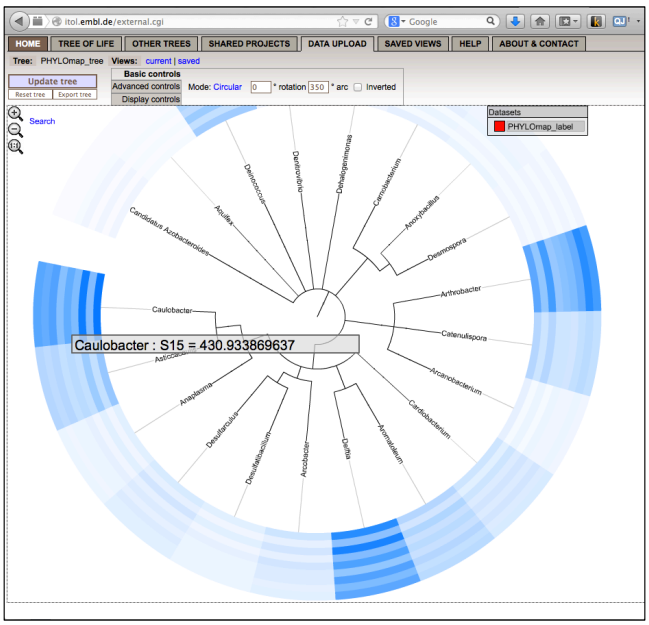


Figure 1: The iTOL's user interface displaying the tree generated from PHYLOmap. One can move/pan the tree, zoom in to get a closer view at a specific part of the heapmap, to display the abundance count and the sample it is from on a mouse-over event, show/hide leaf nodes, prune/collapse/rotate the branches/clades, and change tree display options. Furthermore, the saved views can be shared with other users by simply passing on the generated URL which is kept on the website for a limited period of time.

2 Abundance tables with NCBI taxonomy

To use PHYLOmap, place the software along with the accompanying utilities in the user's home directory at `~/bin` location. The usage information for PHYLOmap is as follows:

```
$ python ~/bin/PHYLOmap.py
Usage:
  python PHYLOmap.py -i <input_file> -o <output_folder> [OPTIONS]

Options:
    -s (--sqrt_transformation)      Apply square-root transformation to the data before generating the
        heatmap (Default: False)
    -w (--heatmap_width) NUM        Width of heatmap in pixels (Default: 20)
    -l (--label) STR                Label of the figure (Default: "PHYLOmap_label")
    -t (--title_tree) STR           Title of the generated tree (Default: "PHYLOmap_tree")
    -x (--min_color) STR            Minimum color in the heatmap (Default: "#FFFFFF")
    -y (--mid_color) STR            Mid color in the heatmap (Default: "#99CCFF")
    -z (--max_color) STR            Maximum color in the heatmap (Default: "#007FFF")
    -r (--tree_file) STR            Newick tree file incase you dont want to use NCBI's taxonomy (Default
        :'')
    --display_mode STR              Tree display mode; "circular" or "normal" (Default: "circular")
    --font_size NUM                 Font size to be used for leaf labels (Default: 10)
    --line_width NUM                Line width in pixels (Default: 1)
    --scale_factor NUM              Default horizontal tree scale will be multiplied with this value (
        Default: 0.5)
$
```

The program accepts the abundance table as a comma-delimited file with sample names as column labels and each row contains the abundance count of a single taxa for different samples, with NCBI scientific names as row labels. The content of an example abundance table in the CSV format is given below:

```
$ cat table.csv
Samples,S10,S11,S12,S13,S14,S15,S16,S17,S18
Anaplasma,17778,5217,15057,5915,10954,8287,9081,6681,7732
Anoxybacillus,8501,2522,6661,3061,8212,5183,4931,4153,4238
Aquifex,3166,1043,3122,1314,2488,1736,2081,1757,1855
Arcanobacterium,10834,4200,9202,4146,8338,7136,6588,8326,7119
Arcobacter,40047,11125,32711,14318,32274,19156,21298,14374,17843
Aromatoleum,80681,33713,96236,45869,74400,43648,80538,52901,69999
Arthrobacter,284594,158455,347003,154873,187129,203726,260902,441160,350738
Asticcacaulis,153936,51937,142155,54750,74398,55042,85625,67303,80820
Candidatus Azobacteroides,6416,1742,5273,1716,7051,2597,2010,1412,1903
```

```
Cardiobacterium,37985,14719,41404,19232,35920,19748,31369,21350,28477
Carnobacterium,12900,3749,9710,4240,13996,8205,5980,5253,6272
Catenulispora,81442,38432,84218,40507,40604,32762,52401,76014,56052
Caulobacter,564129,198191,532423,208191,267620,185704,313987,257678,301914
Dehalogenimonas,4947,2035,5484,2472,3781,2734,3803,3237,3316
Deinococcus,131919,54170,141610,64032,91286,62082,94371,82067,89127
Delftia,442227,170984,496215,193110,324066,166280,313520,201613,281050
Denitrovibrio,6894,2362,6361,2548,4966,3746,4064,3122,3314
Desmospora,6796,2505,6687,3143,5408,3883,4802,4263,4648
Desulfarculus,26624,11289,29870,13776,19292,12192,20433,16735,18862
Desulfatibacillum,14464,5880,15052,6901,11128,7566,10404,7827,9380
$
```

We have chosen a simpler delimited format over the BIOM format [8] for representation of the abundance tables as it is programmatically easy to manipulate these tables within programs as well as from the Linux shell (as can be seen later). The BIOM format is popularised by Qiime [6] as it can store taxonomy as metadata within the file. Even though it offers a structured storage but in most cases of downstream multivariate analyses of the abundance tables (in programs like R), the files are converted to a delimited format before importing them. With the lookup services such as those exported by iTOL, storing redundant information on taxonomy becomes unnecessary. Also, the delimited files can be imported into Microsoft's Excel and can be viewed with ease. Having generated/obtained the abundance table as `table.csv`, one can then use it in PHYLOmap as follows:

```
$ python ~/bin/PHYLOmap.py -i table.csv -o test -s --font_size 100 -w 100
Generated the data file: test/data.csv

Creating the upload params
ncbiIDs : Anaplasma Anoxybacillus Aquifex Arcanobacterium Arcobacter Aromatoleum Arthrobacter
      Asticcacaulis Candidatus_Azobacteroides Cardiobacterium Carnobacterium Catenulispora Caulobacter
      Dehalogenimonas Deinococcus Delftia Denitrovibrio Desmospora Desulfarculus Desulfatibacillum

Uploading NCBI's scientific names to http://itol.embl.de/ncbi_tree_generator.cgi. This may take some
      time depending on how many scientific names are there and how much load there is on the itol server
Generated the tree in newick format: test/data.nwk

Creating the upload params
treeName: PHYLOmap_tree
dataset1MaxPointColor: #007FFF
treeFormat: newick
dataset1MinPointColor: #FFFFFF
dataset1File: test/data.csv
dataset1Label: PHYLOmap_label
dataset1Type: heatmap
dataset1MidPointValue: 391.0
treeFile: test/data.nwk
dataset1Separator: comma
dataset1HeatmapBoxWidth: 100
dataset1MinPointValue: 32.0
dataset1MidPointColor: #99CCFF
dataset1MaxPointValue: 752.0

Uploading the tree to http://itol.embl.de/batch_uploader.cgi. This may take some time depending on how
      large the tree is and how much load there is on the itol server
Tree ID: 1302096432628013960722840
iTOL output: SUCCESS: 1302096432628013960722840

Tree Web Page URL: http://itol.embl.de/external.cgi?tree=1302096432628013960722840&restore_saved=1
SUCCESS: 1302096432628013960722840

Downloading data from http://itol.embl.de/batch_downloader.cgi. This may take some time depending on
      how large the tree is and how much load there is on the itol server

Creating export params
datasetList: dataset1
format: pdf
tree: 1302096432628013960722840
omitDashedLines: 1
displayMode: circular
fontSize: 100
lineWidth: 1
scaleFactor: 0.5

Exported tree to test/data.pdf
$
```

The program displays the parameters used with iTOL's API and generates a PDF document of the phylogenetic tree with the heatmap on it. Furthermore, the tree is also uploaded to iTOL's website and a URL is created. By pasting this URL in the web browser, the tree can be manipulated interactively (see Fig. 1).

3 Abundance tables with any other taxonomy

There are some software that do not support NCBI taxonomy, for example, CREST [2] and RDP [7]. These are widely used bioinformatic programs that perform taxonomic classification of 16S/18S rRNA gene sequences. Instead of using NCBI taxonomy, they use a more controlled and error-fixed taxonomy (SilvaMod, Greengenes), sometimes with groups that are not in NCBI. We can still use them with PHYLOmap by generating the tree in the Newick format from taxonomic paths stored in the assignment files returned from such software, and then providing the tree as an input to the program using `-r` switch.

3.1 Generating data from CREST

The following two commands will produce an assignment file named `input_file_Assignments.txt` for `input_file.fa`:

```
$ megablast -i input_file.fa -b 50 -v 50 -m 7 -d silvamod.fasta -a 10 -o input_file.xml
$ classify -i input_file.fa -a -p input_file.xml
```

To extract an abundance table and the corresponding tree from CREST, the data should first be organized to follow the folder structure shown in Fig. 3a. This organisation makes it easier to run repetitive functionality on the subfolders, each containing an assignment file for a sample with sample names as folder names or what will appear in the abundance table. Moreover, with this structure, we can use the `collateResults.pl` utility that: searches subfolders for files with names matching a certain pattern; extracts the quantitative information stored in a delimited format in them; and then merges the information from all of the subfolders together.

3.1.1 Single Sample

The following bash one-liner will produce an abundance table (`test.csv`) and the corresponding tree in the Newick format (`test.nwk`) at Phylum level for a single sample when run inside the `S1` folder on the terminal (Fig. 3a).

```
$ cat *_Assignments.txt | awk -F"\t" -v k="test.csv" -v l=3 '{a=match($1,"size=.*;");if(a==0) c=1; else
    c=substr($1,RSTART+5,RLENGTH-6);$0=$2";";gsub("\\(","[",$0);gsub("\\)","]",$0);gsub(" ","_",$0);
  split($0,ss,";");$0="";for (i=1;i<=l;i++) $0=$0";"ss[i];gsub("^;","",$0);$0=$0";"; gsub(".*;;$","
  Cellular_organisms;__Unknown__;",$0);gsub("\"","",$0);b[$0]+=c} END {for (i in b) print i;print "
  Sample,S1" > k ;for (i in b) print gensub(".*;(.*?);","\\1","g",i)","b[i] >> k}' | java -classpath
  ~/bin Path2Newick > test.nwk
```

3.1.2 Multiple Samples

The following set of commands will produce an abundance table (`collated.csv`) and the corresponding tree in the Newick format (`collated.nwk`) at Phylum level for multiple samples when run inside the `Main` folder on the terminal (Fig. 3a).

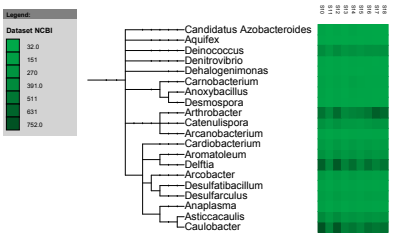
```
$ for i in $(ls -d */); do i=${i%%/}; cd $i; cat *_Assignments.txt | awk -F"\t" -v k="test.csv" -v l=3
  '{a=match($1,"size=.*;");if(a==0) c=1; else c=substr($1,RSTART+5,RLENGTH-6);$0=$2";";gsub("\\(","["
  ,$0);gsub("\\)","]",$0);gsub(" ","_",$0);split($0,ss,";");$0="";for (i=1;i<=l;i++) $0=$0";"ss[i];
  gsub("^;","",$0);$0=$0";"; gsub(".*;;$","Cellular_organisms;__Unknown__;",$0);gsub("\"","",$0);b[$0
  ]+=c} END {for (i in b) print i;for (i in b) print gensub(".*;(.*?);","\\1","g",i)","b[i] >> k}';
  cd ..; done | java -classpath ~/bin Path2Newick > collated.nwk
$ perl ~/bin/collateResults.pl -f . -p test.csv > collated.csv
```

The above bash one-liner generates a `test.csv` file in each subfolder, which is then used in the second command to collate the results together to produce `collated.csv`.

3.2 Generating data from RDP

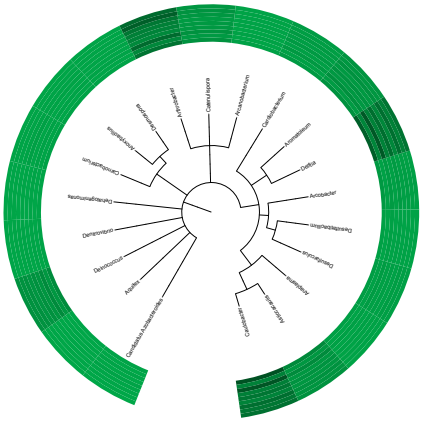
The following command will produce an output file named `input_file_Assignments.txt` for `input_file.fa`:

```
$ java -Xmx1g -jar classifier.jar classify -f filterbyconf -o input_file_Assignments.txt input_file.fa
```



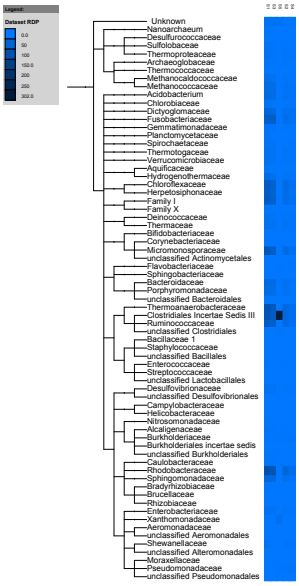
(a) NCBI normal

```
python ~/bin/PHYLOmap.py -i table.csv -o test -s
-w 15 -l NCBI -t NCBI_tree --display_mode normal
--font_size 20 --line_width 1 --scale_factor 0.1 -x
"#00A849" -y "#008A3C" -z "#005223"
```



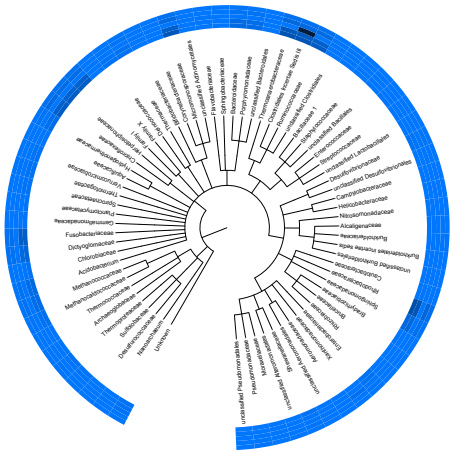
(b) NCBI circular

```
python ~/bin/PHYLOmap.py -i table.csv -o test -s -w
15 -l NCBI -t NCBI_tree --font_size 20 --line_width
1 --scale_factor 0.2 -x "#00A849" -y "#008A3C" -z
"#005223"
```



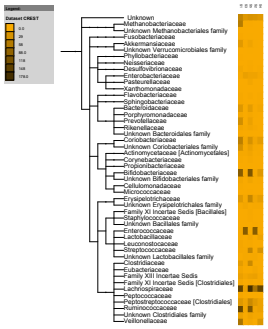
(c) RDP normal (Family level)

```
python ~/bin/PHYLOmap.py -i collated.csv -o collated
-s -w 15 -l RDP -t RDP_tree -r collated.nwk
--display_mode normal --font_size 20 --line_width
1 --scale_factor 0.1 -x "#0078FF" -y "#003E85" -z
"#001C3B"
```



(d) RDP circular (Family level)

```
python ./~/bin/PHYLOmap.py -i collated.csv
-o collated -s -w 15 -l RDP -t RDP_tree -r
collated.nwk --font_size 20 --line_width 1
--scale_factor 0.2 -x "#0078FF" -y "#003E85" -z
"#001C3B"
```



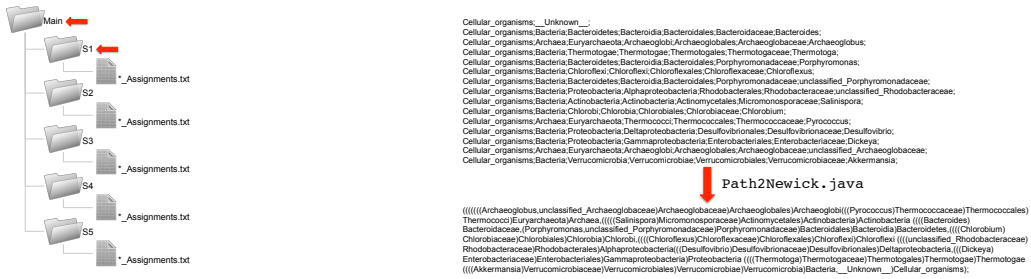
(e) CREST normal (Family level)

```
python ~/bin/PHYLOmap.py -i collated.csv -o collated
-s -w 15 -l CREST -t CREST_tree -r collated.nwk
--display_mode normal --font_size 20 --line_width
1 --scale_factor 0.1 -x "#FAAB00" -y "#916300" -z
"#452F00"
```

(f) CREST circular (Family level)

```
python ~/bin/PHYLOmap.py -i collated.csv -o collated
-s -w 15 -l CREST -t CREST_tree -r collated.nwk
--font_size 20 --line_width 1 --scale_factor 0.2 -x
"#FAAB00" -y "#916300" -z "#452F00"
```

Figure 2: Output produced by PHYLOmap for example datasets along with the parameters used in the program.



(a) Folder structure for the correct execution of `bash` one-liners

(b) `Path2Newick` functionality

Figure 3: The `bash` one-liners mentioned in this document can be used to generate input data for `PHYLOmap` for both single (in the `S1` folder) as well as multiple samples (in the `Main` folder). These `bash` one-liners rely on `Path2Newick` (3b) utility to convert extracted paths to tree in the Newick format.

3.2.1 Single Sample

We follow the same folder structure as mentioned in the previous section. The `bash` one-liners for `RDP` are slightly different due to the different format of the assignment files, however they follow the same philosophy. The following `bash` one-liner will produce an abundance table (`test.csv`) and the corresponding tree in the Newick format (`test.nwk`) at Phylum level for a single sample when run inside the `S1` folder on the terminal (Fig. 3a).

```
$ cat *_Assignments.txt | awk -F"\t" -v k="test.csv" -v l=3 '{a=match($1,"size=.*;");if(a==0) c=1; else
    c=substr($1,RSTART+5,RLENGTH-6);$0="Cellular_organisms;"$3";"$6";"$9";"$12";"$15";"$18";";gsub("
    \\(","[",$0);gsub("\\)","]",$0);gsub(" ","_",$0);split($0,ss,";");$0="";for (i=1;i<=l;i++) $0=$0";"
    ss[i];gsub("^;","",$0);$0=$0";"; gsub(".*;;$","Cellular_organisms;__Unknown__;",$0);gsub("\"","",$0
    );b[$0]+=c} END {for (i in b) print i;print "Sample,S1" > k ;for (i in b) print gensub(".*;(.*?);",
    "\\1","g",i)","b[i] >> k}' | java -classpath ~/bin Path2Newick > test.nwk
```

3.2.2 Multiple Samples

The following set of commands will produce an abundance table (`collated.csv`) and the corresponding tree in the Newick format (`collated.nwk`) at Phylum level for multiple samples when run inside the `Main` folder on the terminal (Fig. 3a).

```
$ for i in $(ls -d */); do i=${i%%/}; cd $i; cat *_Assignments.txt | awk -F"\t" -v k="test.csv" -v l=3
    '{a=match($1,"size=.*;");if(a==0) c=1; else c=substr($1,RSTART+5,RLENGTH-6);$0="Cellular_organisms;
    "$3";"$6";"$9";"$12";"$15";"$18";";gsub("\\(","[",$0);gsub("\\)","]",$0);gsub(" ","_",$0);split($0,
    ss,";");$0="";for (i=1;i<=l;i++) $0=$0";"ss[i];gsub("^;","",$0);$0=$0";"; gsub(".*;;$","
    Cellular_organisms;__Unknown__;",$0);gsub("\"","",$0);b[$0]+=c} END {for (i in b) print i;for (i in
     b) print gensub(".*;(.*?);","\\1","g",i)","b[i] >> k}'; cd ..; done | java -classpath ~/bin
    Path2Newick > collated.nwk
$ perl ~/bin/collateResults.pl -f . -p test.csv > collated.csv
```

For producing trees at other taxonomic levels, in the first `awk` statement in the above `bash` one-liners, use: `l=4` for Class, `l=5` for Order, `l=6` for Family, and `l=7` for Genus, respectively. These `bash` one-liners will work even if we run the classifiers on the dereplicated reads in the `usearch` header format i.e. containing the string `size=.*` in the FASTA headers. If the original dereplicated sequences are in a different format, we first convert them to the `usearch` header format before running the classifiers and using the above `bash` one-liners. For example, if we have generated the denoised and dereplicated reads from `AmpliconNoise` [1], then the following `bash` one-liner can be used to convert the FASTA headers:

```
$ awk '/>/{$0=gensub("(.*?)_(.*?)$","\\1;size=\\2;","g",$0)}1' dereplicated.fa >
    dereplicated_usearch_format.fa
```

In the above `bash` one-liners, we place any unassigned read at a certain taxonomic level as `__Unknown__` to `Cellular_organisms` as well as fix errors in generated paths such as inverted commas, replace spaces with underscores, and parentheses with square brackets, respectively.

4 Conclusion

We have written a general purpose utility that generates the phylogenetic tree with a heapmap for the taxa present in the abundance tables and can integrate with majority of the taxonomic assignment software used for metagenomic analyses. All the figures in this document can be reproduced by running the example datasets through PHYLOmap with the parameters mentioned in Fig. 2 and the bash one-liners mentioned in the document.

References

[1] AmpliconNoise. https://code.google.com/p/ampliconnoise/.

[2] CREST (LCAClassifier). http://code.google.com/p/lcaclassifier/.

[3] Earth Microbiome Project. http://www.earthmicrobiome.org/.

[4] iTOL Interactive Tree of Life. http://itol.embl.de.

[5] MEGAN 4 - MEtaGenome ANalyzer. http://ab.inf.uni-tuebingen.de/software/megan/.

[6] Qiime Quantitative Insights into Microbial Ecology. http://qiime.org/.

[7] RDP Classifier. http://sourceforge.net/projects/rdp-classifier/files/.

[8] The Biological Observation Matrix (BIOM) format. http://biom-format.org/.