User's guide: Manual for Metaxa2.1.2

This is a guide on how to install and use the software utility Metaxa2. The software is written for Unix-like platforms, and should work on nearly all Linux-based systems, as well as MacOS X.

Contents of this manual

1. Detailed installation instructions

The README.txt file bundled with the script provides a quick installation guide.

In order to install certain packages, you might need to have superuser privileges. For installation on Mac, you will have to install the Apple Xcode package available on your MacOS X System DVD (or online in the latest MacOS X releases) in order to be able to compile programs. Please talk to your system administrator if you feel unsure about these steps. Note that the packages are mandatory and that you should not proceed unless these criteria are fulfilled.

[If you don't have superuser privileges on your machine: Create a directory within your user directory, e.g. /home/user/bin/, and to store all required binaries there. By adding this directory to your PATH, any software placed in the directory will behave as if installed for all users using superuser privileges. If you use the bash shell, you can add a bin directory to your PATH, by adding the line "export PATH=$PATH:$HOME/bin/:." to the file .profile in your home directory. The process of adding items to one's PATH varies among systems and shells. Close the terminal and open a new one for this change to take effect.]

Perl needs to be installed on the computer. Most Unix-based systems including Linux and MacOS X have Perl pre-installed. You can check this by opening a command line terminal and type "perl -v". In case Perl is not installed, you have to download (http://www.perl.org) and compile the program.

Download and install HMMER version 3 (http://hmmer.janelia.org/software). Version 2 of Metaxa relies on HMMER version 3, just as the previous Metaxa versions. Metaxa2 will **_not_** work with earlier versions of HMMER, although it will work with the HMMER 3.1 beta. Download the HMMER package source code to your preferred directory such as /home/user/. Open a command line terminal, move into the directory with "cd /home/user/" and unpack the tarball with "tar -xvfz hmmer-3.0.tar.gz". Now, you need to

compile HMMER from source files. To compile it from source, enter the new directory and follow the installation instructions in the file INSTALL.

If you have trouble compiling HMMER, you can try to use the pre-compiled binaries available at the HMMER home page. After download and unpacking of the tarball, the binaries are located in the binaries directory contained within the newly created HMMER directory. Move into the binaries directory and move all of its contained files into your preferred bin directory (usually either /usr/local/bin/ or your own bin directory, /home/user/bin/). The HMMER package should now be installed on your computer; you can check this by typing "hmmscan -h" in the terminal and press enter; you should now see HMMER output.

Download and install the BLAST package (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/ release/LATEST/) for sequence similarity searches. Metaxa2 works with both the legacy version of BLAST, *as well as* BLAST+. It should work with any version of BLAST starting with version 2.2 and later. Download the BLAST package for your operating system to your preferred directory. Open a command line terminal, move into the directory with "cd /home/user/" and unpack the tarball with "tar -xvfz blast-2.2.24-platform.tar.gz". Move into the bin directory inside the newly created BLAST directory, and move all of its contained files into your preferred bin directory. Alternatively, you can add the BLAST bin directory to your PATH. The BLAST package should now be installed on your computer; you can check this by typing "blastall -" in the terminal and press enter; you should now see the listing of BLAST options.

Download and install the MAFFT (http://mafft.cbrc.jp/alignment/software/) for multiple alignment. Metaxa2, as previous versions of Metaxa, relies on MAFFT version 6. MAFFT is not critical for Metaxa2's core functions, but is used for automatically creating alignments of uncertain sequences. Instructions for installing MAFFT are available on the MAFFT download page.

Go to http://microbiology.se/software/metaxa2 in order to download the Metaxa2 package. Download it to your preferred directory. Unpack the downloaded tarball with "tar -xvfz metaxa2_2.0.tar.gz". A directory called Metaxa2 will be created. You will see a number of files and directories inside it, including metaxa2, metaxa2_x, metaxa2_c, install_metaxa2, and the metaxa2_db directory (containing the Hidden Markov Models and a BLAST database), the user's guide, and test input files. Enter the directory, and type "./install_metaxa2". Press enter and follow the on-screen instructions. You will be prompted for whether you have superuser privileges and for the location where you want Metaxa2 to be installed. If Metaxa2 is successfully installed you should see its help message when typing the command "metaxa2 --help".

2. Changes from previous versions of Metaxa

Version 2.1 of Metaxa2 differs from the previous version in a few ways, outlined in the first paragraph below. In addition, Metaxa2 introduces a couple of new features over Metaxa 1.1.2, as well as changing the default behavior of some options. The main changes are outlined in the following paragraphs.

Changes in Metaxa2 version 2.1

Metaxa2 version 2.1 incorporates a number of minor changes and improvements. These include a new detection mode for detection of multiple rRNA sequences in e.g. full genomes, an option to specify reference sequences to exclude from the analysis, to be able to sort out specific non-target sequences from the dataset (e.g. from a host organism), the possibility to get separate files for paired-end reads matching rRNA for further downstream analysis, and the important addition of the Metaxa2 Diversity Tools (see section 6, further below).

The new genome mode allows Metaxa2 to be used to find multiple rRNA sequences in longer stretches of DNA, such as complete genomes or contigs. It also comes with an automatic option, which processes sequences longer than 2500 bases in the genome mode and shorter sequences in the metagenomic mode.

The new reference option can be used to supply Metaxa2 with a FASTA file with reference sequences that should *not* be included in the output. These sequences can, for example, be the rRNA sequences from a host organism, or from some particular taxon that the user is not interested in studying.

Finally, the default option for the "--align" flag has been changed to "none", since with increasing data set sizes the automatic alignment became a very time-consuming step. To use this capability as present by default in previous versions of Metaxa, add the "--align auto" option to the command line. Metaxa2 2.1 is also able to automatically detect whether legacy BLAST or BLAST+ is installed and will (normally) be able to choose which one to use. Still, the "--plus" option can be used to prefer one over the other.

Extraction and classification of LSU rRNA sequences

One of the major new features in Metaxa2 is the addition of a second, frequently used gene in addition to the SSU gene: the large subunit rRNA (LSU), also known as 23S rRNA in prokaryotes and 25S or 28S rRNA in eukaryotes. Toggling the switch from the SSU gene to the LSU gene is done using the "-g lsu" or "--gene lsu" options (both set to "ssu" by default). The operation of Metaxa2 is then the same as when searching for and extracting SSU genes. The databases of Metaxa2 representing the SSU and LSU genes are kept separately within the metaxa2_db directory. Although the support of Metaxa2 for LSU extractions is reliable and robust, it has not been as extensively tested internally as the SSU extractions. Therefore, we encourage users to report suspicious – or obviously misclassified – entries, so that the LSU support of Metaxa2 can be improved even further in the future.

Completely redesigned system for taxonomic classifications

One of the design goals of Metaxa2 was to be able to make more sensible predictions of the origin of each input sequence identified as an SSU or LSU gene, even at very short read lengths. To achieve this, we have completely rewritten the classification engine, which has enabled Metaxa2 to produce reliable predictions of the taxonomic affiliation of the extracted SSU/LSU genes. Metaxa2 reports those affiliations to a file with the suffix ".taxonomy.txt". By using the taxonomic data of the best five BLAST matches (by default), Metaxa2 calculates the most accurate taxonomic placement possible for the input sequence, given the affiliations

of the matching hits, and their degree of identity to the input sequence. Each entry is associated with a taxonomic affiliation/level, and three numbers; the percent identity to the best matching BLAST hit, the length of that alignment, and a reliability score. The reliability score is calculated based on the percent identity to the best BLAST hit, and how divergent the rest of the BLAST hits are from the first one. The maximum value of this score is 100. The score of 100 is only given if *all five* BLAST matches are 100% identical to the input sequence, and all those matches represent the same taxonomic lineage. This means that the situation outlined above rarely – if ever – occurs on real data. Instead, scores above 80 should be considered trustworthy. The reliability score can then be used to filter out uncertain entries when summarizing the taxonomic predictions with the Metaxa2 Taxonomic Traversal Tool (metaxa2_ttt), described below.

The Metaxa2 Taxonomic Traversal Tool – metaxa2_ttt

Because of the extended abilities of Metaxa2 to classify SSU and LSU sequences in greater detail, a new tool to investigate the organismal content of the sample at different taxonomic levels has been included in the Metaxa2 package. This tool is called the Metaxa2 Taxonomic Traversal Tool, since it does exactly that – it traverses the ".taxonomy.txt" output file and reports Metaxa's taxonomic predictions according to specified cutoffs. The complete usage of the metaxa2_ttt tool is described in part five of this manual (*Metaxa2 Taxonomic Traversal Tool*). In short, the traversal tool outputs the number of identified SSU/LSU sequences associated with each node in the taxonomic tree, at different levels (roughly corresponding to kingdoms, phyla, classes, orders, families, genera, species, subspecies, etc.)

The "--guess_species" option is now obsolete

Since Metaxa2 now handles taxonomy in a much more elaborate way, the "--guess_species" option has been deemed obsolete. It is, however, still possible to use this option, but a warning message will appear. The function may be deprecated in future versions of Metaxa, and is not actively maintained anymore (and has thus not been tested in the Metaxa2 evaluation phase).

Updated database for classification

The improved classification engine has naturally required us to update and improve upon the underlying databases. The basis of the Metaxa2 databases is the SILVA reference release 111 and Mitozoa release 10. From this data, we have created a curated reference database, in large part by automated means, but also using extensive manual curation and cross-checking in GreenGenes, CRW and GenBank. Suspicious or erroneous entries were removed, as were sequences from uncultured or unverified organisms.

Direct input of sequences in FASTQ-format

Metaxa2 now supports input of sequence data sets directly in the FASTQ format. Although Metaxa2 will try to auto-detect the format of the input file, specifying "-f fastq" will force Metaxa2 to read input in FASTQ format. This option might be particularly useful when piping input to Metaxa2 using standard input. Note that FASTQ files are only supported when Metaxa2 is run in pipeline mode, i.e. not when the two tools metaxa2_x and metaxa2_c are run separately. These two tools still expect input in FASTA format.

Support for paired-end libraries

In addition to FASTQ support, Metaxa2 also handles paired-end libraries. Paired-end data needs to be supplied in two separate files, with sequence 1 in both files corresponding to the two ends of the same read, and so on. If your data is held in a single file, a utility such as the pefcon and pesort tools of the PETKit (http://microbiology.se/software/petkit/) can help you. To enable Metaxa2's paired-end capabilities, use the options "-1 firstfile -2 secondfile" instead of the "-i" option for input. Metaxa2 will automatically orient the reads in the same direction, and utilize a combination of the two ends for the extraction and classification process. Note that the two ends will be output ***together*** as one single read with a spacer, unless the "--split_pairs" option is used.

Improved support for libraries with short read lengths (~100 bp)

Since metagenomics is moving towards shorter read lengths and larger data sets (e.g. Illumina and IonTorrent sequencing), special care has been taken to try to address the short-read issues in previous Metaxa versions. We have tried to optimize the default options for reads down to ~100 bp. This will, however, increase the likelihood of false positive extractions somewhat. Moreover, the optimization towards shorter reads should be kept in mind when using Metaxa2 on longer reads, which might require tweaking of the default options (one may, for example, compare to the default options of Metaxa 1.1.2).

Quality pre-filtering of reads in FASTQ-format

Metaxa2 can use the quality information in the FASTQ files to filter out low-quality reads or read pairs. This is controlled by a range of options described in part three of the manual (*Usage and Commands*).

Support for the modern BLAST+ package

Metaxa2 now incorporates the long sought-after support for the NCBI BLAST+ package, enabling users to move on to what is deemed the future of BLAST. In addition, the software can select BLAST version automatically since Metaxa2 version 2.1.

Compatibility with HMMER 3.1

Not only does Metaxa2 bring BLAST+ compatibility, it also secures support for the upcoming version of HMMER (released in beta at the time of writing).

Changed default priority for scoring best HMMER match

As part of the optimization for larger data sets and shorter reads, Metaxa's priority system for determining the best HMMER match has now been changed from "sum" to "score". To switch back to Metaxa 1.X behavior, one can use the "--selection_priority sum" option. Be aware of that this might influence other parameters as well!

HMMER's heuristics are now used by default

Another optimization for larger data sets is that the heuristics of HMMER has now been turned on by default. To switch the heuristics of, use the "--heuristics F" option. Note, however, that this will make HMMER *much* slower on large data sets.

3. Usage and commands

For the *very* impatient only: follow the brief installation instructions in the file README.txt. To check for SSU rRNA sequences in the file test.fasta, you would then type "metaxa2 -i test.fasta -o test" on the command line. To check for LSU rRNA sequences, type "metaxa2 -i test.fasta -o test -g lsu" instead.

For all other users: Metaxa2 accepts input in the FASTA and FASTQ formats. As it pre-processes the input sequences it is possible to input both aligned and unaligned FASTA files, containing both DNA and RNA sequences. By default, Metaxa2 outputs ten files; one summary file of the entire run, one "graphical" representation of hits, one FASTA file of all identified SSU sequences, one FASTA file for each of the six possible origins, and one file containing the predicted taxonomic origin of the extracted sequences. To list all the available options for Metaxa2, type "metaxa2 --help". You can use the test.fasta file that comes bundled with the software for a test run. This file contains 50 randomly selected SSU entries, ten of each origin; 50 randomly selected LSU entries and 10 non-SSU, non-LSU sequences. In the simplest case, Metaxa2 is run by "metaxa2 -i input_file -o output". Below is a listing of all options Metaxa2 accepts. Boolean options can be turned on with "T", "true" or "1" and off using "F", "false" or "0".

Main options:

-i {file}	Nucleotide FASTA/FASTQ input file to investigate. Metaxa2 accepts both aligned and unaligned FASTA. If no input is specified, Metaxa2 will read sequences from standard input, which means that FASTA sequences can be piped into Metaxa2.
-o {file}	Base for the file names of the output files. Suffixes will be added automatically. Defaults to metaxa_out.
-1 {file}	DNA FASTA/FASTQ input file containing the first reads in the read pairs to investigate. Use instead of the -i option.
-2 {file}	DNA FASTA/FASTQ input file containing the *second* reads in the read pairs to investigate. Use instead of the -i option, and only together with the -1 option. Note that the reads in the files must be in the same order.
--pairfile {file}	As an alternative to using the -1 and -2 options, the user might use -i and --pairfile instead, which has the same meaning.
-f {a, auto, f, fasta, q, fastq, p, paired-end, pa, paired-fasta}	Specifies the format of the input file(s). By default, Metaxa2 will try to auto-detect the format (auto). Synonymous to "--format".
-g {ssu, lsu}	Specifies if Metaxa2 should identify and extract SSU or LSU rRNA genes. By default, Metaxa2 will look for SSU genes (ssu).
--mode {m, metagenome, g, genome, a, auto}	Controls the Metaxa2 operating mode. Genome (and auto) mode allows the detection of multiple matches on the same DNA sequence. Default is the 'metagenome' mode.
-p {directory}	A path to a directory containing HMM-profile collections representing SSU rRNA conserved regions. By default, Metaxa assumes to find the databases in the metaxa_db directory, located in the same directory as Metaxa itself.
-d {database}	The BLAST database used for classification. By default, Metaxa assumes to find the databases in the metaxa_db directory, located in the same directory as Metaxa itself.

--date {T or F}	Adds a date and time stamp to the output file. This can be useful e.g. if Metaxa is part of a pipeline where input files with the same name could cause overwriting of important data. Off (F) by default.
--plus {T or F}	If enabled, the BLAST search will (preferentially) be performed using BLAST+ instead of the legacy blastall engine. By default, blastall is used (F).

FASTQ and Paired-end specific options:

Note that FASTQ format is only supported when Metaxa2 is run in pipeline mode.

-q {value}	Minimum quality value for a nucleotide to be considered "good". Default = 20.
--quality_percent {value}	Percentage of low-quality (below -q value) accepted before filtering/trimming is started. Default=10.
--quality_filter {T or F}	If enabled, Metaxa2 will filter out low-quality reads (below specified -q value), removing them entirely from the input data before running the HMMER searches. Off (F) by default.
--quality_trim {T or F}	If enabled, Metaxa2 will trim bad bases (below -q value) from the end of the read. This is generally to prefer over filtering the read entirely. Off (F) by default.
--ignore_paired_read {T or F}	When enabled, Metaxa2 will not discard the entire read pair if only one of the reads is of bad quality and would be filtered out. On (T) by default.
--distance {value}	Specifies the distance between the sequence pairs (often referred to as the insert size). Changing this value has little impact on Metaxa2's actual performance. Default = 150.

Sequence selection options:

-t {b, bacteria, a, archaea, e, eukaryota, m, mitochondrial, c, chloroplast, A, all}	Set of profiles to use for the search (comma-separated). Accepts any list of sets, e.g. "bacteria,chloroplast", "m,c" or "eukaryota". Can be used to restrict the search to only a few SSU/LSU types to save time, if one or more of the origins are not relevant to the dataset under study. Default is to use all (the "all" option).
-E {value}	Domain E-value cutoff a sequence must obtain in the HMMER-based step to be included in the output. Default = 1.
-S {value}	Domain score cutoff for a sequence must obtain in the HMMER-based step to be included in the output. Default = 12.
-N {value}	The minimum number of domains that must match a sequence for it to be included in the output. Setting the value lower than two will increase the number of false positives, while increasing it above two will decrease Metaxa's detection abilities on fragmentary data. Default = 2.
-M {value}	Number of top BLAST matches that should be considered in classification. This setting also affects the number of matches used for taxonomic classification. Default = 5.
-R {value}	Reliability score cutoff for the taxonomic classification. Entries not satisfying the cutoff will be classified at the level above in the taxonomic hierarchy, until the reliability score is above the threshold. Default = 80.
-T {comma-separated values}	Sets the percent identity cutoff to be classified at a certain taxonomic level. If the percent identity to a sequence in the Metaxa2 database is below this cutoff, the sequence will not be classified at that taxonomic

	resolution. The order of the values is: Kingdom/Domain,Phylum,Class,Order,Family,Genus,Species Default: 0,60,70,75,85,90,97
-H {value}	The number of points that the predicted origin of the Metaxa Extractor is given. Default is the same as the number of sequences used for classification (-M option above), which is set to 5 by default.
--selection_priority {score, sum, domains, eval}	Determines what will be of highest priority when assessing the origin of the sequence. Options are:

- score, which uses the average score of the found hits

- sum, which sums the scores for each profile match and divides the sum by the number of profiles of the given type

- domains, which uses the number of domains retrieved of a given type

- eval, which uses the average E-value of the found hits

Default is to use score. |
--search_eval {value}	The actual E-value cutoff used in the HMMER search. High numbers may slow down the process. Should never be set to a lower value than the -E option. Cannot be used in combination with the --search_score option. Default is to use score cutoff (see --search_score below), not E-value.
--search_score {value}	The score cutoff used in the HMMER search. Low numbers may slow down the process. Should never be set to a higher number than the -S option. Cannot be used in combination with the --search_eval option. Default = 0.
--blast_eval {value}	The E-value cutoff used in the BLAST search. High numbers may slow down the process. Cannot be used in combination with the --blast_score option. Default is 1e-15.
--blast_score {value}	The score cutoff used in the BLAST search. Low numbers may slow down the process. Cannot be used in combination with the --blast_eval option. Default is to use E-value cutoff (see --blast_eval above), not score.
--blast_wordsize {value}	The word-size used for the BLAST-based classification. Lower numbers will slow down the process significantly, while higher numbers may potentially decrease classification accuracy. Default is 14.
--allow_single_domain {e-value,score or F}	Allow inclusion of sequences that only find a single domain, given that they meet the more stringent E-value and score thresholds specified. By default, single domains are allowed, with E-value cutoff 1e-10 and score cutoff 0 ("1e-10,0").
--allow_reorder {T or F}	Allows profiles not to be in the expected order (1-9) on the extracted sequences. If turned off, a file of potential chimeric sequences (with profile matches in the wrong order) is written, allowing for rudimentary chimera detection. This can (and should) be used on full-length sequences. On fragmented sequences, however, the risk of missing true positives increases if this option is turned off. On (T) by default.
--complement {T or F}	If on, Metaxa2 checks both DNA strands for matches to HMM-profiles. On (T) by default.
--cpu {value}	The number of CPU threads to use. Metaxa performs significantly faster using more CPUs. Default is 1.
--multi_thread {T or F}	Multi-thread the HMMER-search. On (T) by default if the number of CPUs is larger than one (--cpu option > 1), else off (F).

--heuristics {T or F}	Selects whether to use HMMER's heuristic filtering. On (T) by default. Turning this setting off will decrease speed, but increase precision.
--megablast {T or F}	Uses megablast for classification for better speed but less accuracy. Off (F) by default.
--reference {file}	Enables to sort out sequences from a file of reference entries in FASTA format. Input sequences matching the reference sequences will be sent to a separate file in the analysis. Default is that this option is not used.
--ref_identity {value}	The sequence percent identity cutoff for a sequence to be considered being derived from a reference entry. Default is 99.

Output options:

--summary {T or F}	If on, Metaxa outputs a summary of results. File suffix is ".summary.txt". On (T) by default.
--graphical {T or F}	If on, Metaxa outputs "graphical" text representations of where in each sequence the conserved domains were found. File suffix is ".graph". On (T) by default.
--fasta {T or F}	If on, FASTA-formatted files containing the extracted SSU sequences are written. One file for each origin is written, plus an extraction file containing all SSUs identified in the first analysis step. On (T) by default.
--split_pairs {T or F}	Outputs the two read pairs in two separate files instead of as a joint rRNA sequence. Off (F) by default.
--table {T or F}	If on, Metaxa saves table format output of results, separately for HMMER and BLAST output. Note that neither of these outputs is the *actual* output of the respective program. To get these file, use the "--save_raw T" (see below). Off (F) by default.
--taxonomy {T or F}	Table format output of probable taxonomic origins for the identified SSU/LSU sequences. File suffix is ".taxonomy.txt". On (T) by default.
--taxlevel {integer}	Forces Metaxa to classify sequences at a certain taxonomy level, regardless of their reliability score. Off (0) by default.
--not_found {T or F}	If on, Metaxa outputs a list of entries that do *not* seem to be SSU sequences. File suffix is "_not_found.txt". Off (F) by default.
--align {a, all, u, uncertain, n, none}	Outputs alignments of BLAST matches to each query in all (a), uncertain (u) or no (n) cases. Requires MAFFT to be installed. Default is to output alignments in uncertain cases (u).
--truncate {T or F}	Removes ends of SSU sequences if they are outside of the SSU region. If off, the whole input sequence is saved. On (T) by default.
--guess_species {T or F}	Deprecated option, use the --taxonomy option instead. Kept for compatibility with previous Metaxa versions. Off (F) by default.
--silent {T or F}	Suppresses printing of progress info to screen. Off (F) by default.
--graph_scale {value}	Sets the scale of the graphical output. If the provided value is zero, a percentage view is shown. Default is 0.
--save_raw {T or F}	Saves all raw data for searches etc. instead of removing it when finished. Saves data to a directory with the suffix "_metaxa_raw_output". Off (F) by default.

Information options:

-h	Displays basic usage help.
--help	Displays the help message, complete with all options.
--bugs	Displays the bug fixes and known bugs in this version of Metaxa.
--license	Displays licensing information.

4. Output files

Metaxa2 outputs a number of files, depending on the selections of the user (see Usage and Commands above). By default, seven FASTA-files, a table of taxonomic classifications, a file containing graphical representation of putative SSU/LSU sequences, and a summary file is written. In addition, tables of BLAST and HMMER results, lists of non-targeted entries, and sequence alignments can be written on request by the user. There is also an option to preserve all the intermediate data generated by the HMMER and BLAST searches.

FASTA-output

Metaxa2 generates one FASTA file for each origin (archaea, bacteria, eukaryota, chloroplast, and mitochondria), one file containing sequences of uncertain origin, and one file with all rRNA sequences identified and extracted in the first step. Sequences in these files are marked according to their origin. Sequences whose origin Metaxa2 could not establish with certainty, but for which enough data were available to allow a qualified guess as to the origin of the sequences, are marked with a "#" character at the end of the definition line. A certain sequence may look like this:

```
>gi|117927211 Bacterial 16S SSU rRNA
GTTTGATCCTGGCTCAGGACGAACGCTGGCGGCGTGCTTAACACATGCAAGTCGAGCGGA…
```

Note that Metaxa2 has added the type of the SSU sequence ("Bacterial 16S SSU rRNA") to the definition line in the example above. An uncertain sequence could look like this:

```
>AABL01000014.4508.5931 Putative Chloroplast 16S SSU rRNA #
GAACGCTAGAAATATACATTACACATGCAAATTTATGATAATATCATAGTGAATAGGTGA…
```

The extraction file contains all sequences identified as rRNA by metaxa2_x (the first step of the analysis). The sequence entries in that file contain information on the rRNA domains that were found in each sequence and what origin that is most likely based on the profile search. An example is shown below:

```
>A16379.1.1496|B Predicted Bacterial 16S SSU rRNA (1447 bp) From
domain V1l to V9r on main strand Found domains: V1l V2l V2r V3l V3r
V4l V4r V5l V5r V6l V7l V8l V8r V9l V9r
CAGGCTTAACACATGCAAGTCGAACGGTAGCACGAAGGACTTGCTCCTTGGGTGACGAGT…
```

Summary

A summary of the Metaxa2 run is written to a file with the suffix ".summary.txt". In this file the statistics of the run is collected, as are the starting and ending times for the run. Also, lists of the identifiers of extracted SSU sequences are written to this file, one list for each origin. The first section of the file shows the data from the extraction step. The second section is associated with the second classification step. After the second section, the lists of entries of different origins are found. An example of parts of a summary file is shown below:

```
Metaxa run started at Tue Jul 23 10:07:52 2013.
------------------------------------------------------------------
Number of sequences in input file:        100
Sequences detected as SSU rRNA by Metaxa: 100
  On main strand:           91
  On complementary strand:   9
SSU sequences by preliminary origin:
  Archaea:          0
  Bacteria:         0
  Eukaryota:        0
  Chloroplast:    100
  Mitochondria:     0
  Other:            0
------------------------------------------------------------------
Number of SSU rRNA sequences to be classified by Metaxa:    100
Number of SSU rRNA having at least one database match:      100
Number of SSU rRNA successfully classified by Metaxa:       100
Number of uncertain classifications of SSU rRNA sequences:  0
Total number of classifications made by Metaxa:             100
Number of SSU rRNA sequences assigned to each origin:
  Archaea:          0
  Bacteria:         0
  Eukaryota:        0
  Chloroplast:    100
  Mitochondria:     0
  Uncertain:        0
------------------------------------------------------------------
Sequences of archaeal origin (16S):
------------------------------------------------------------------
Sequences of bacterial origin (16S):
------------------------------------------------------------------
Sequences of eukaryote origin (18S):
------------------------------------------------------------------
Sequences of chloroplast origin (16S):
Acorus_americanus_AcamCr001
Aethionema_cordifolium_AecoCr001
...
Welwitschia_mirabilis_WemiC_r001
Zea_mays_ZemaCr113
------------------------------------------------------------------
Sequences of mitochondrial origin (12S and 16S):
------------------------------------------------------------------
Sequences of uncertain origin:
------------------------------------------------------------------
Metaxa run finished at Tue Jul 14 10:08:42 2013.
```

Taxonomy table

One of the new features of Metaxa2 is the substantially improved ability to make taxonomic predictions of identified rRNA sequences. The results of these predictions are written to a file with the suffix ".taxonomy.txt". Each input sequence is represented by one line in this tab-separated file with five columns.

Column	Description
ID	The identifier of the query sequence.
Classification	The taxonomic tree for which Metaxa2 has been able to a reliable prediction.
Identity	The percent identity to the best BLAST match in the database.

Length	The length of the alignment of the input sequence and the best BLAST match.
Reliability score	The number of conserved domains for the most likely origin that was found in the sequence. The reliability score is calculated based on the percent identity to the best BLAST hit, and how divergent the rest of the BLAST hits are from the first one. The maximal score is 100, and the minimum score is determined by the -R cutoff used (80 by default). Scores above 80 can generally be considered good.

Graphical representations

Metaxa2 writes graphical (ASCII) representations of where in each sequence the various conserved regions were found to a text file with the suffix ".graph". Separate graphs are written for each origin and strand, which means that each sequence entry may be present more than once in this file, if it has matches to HMM-profiles from more than one origin. This makes it possible to manually inspect how Metaxa2 has evaluated each sequence. The graphical representations look like this:

```
B matches on main strand:
>> id|454_30|gi|50402825|gb|AY687385.1|   403 bp
------------------V5l=========V5r===============================
****************************************************************
```

The first row shows the type of the entries below, as well as the strand they are found on. Each entry begins with the characters ">> ", followed by the sequence identifier, and its length. Below the identifier row, the sequence graph is shown. By default, all sequences are scaled so that they are of equal length, and the domains are placed according to their *relative* position in the sequence. The characters that are used in the graphical representations are explained in the table below.

Feature	Description
-	Part of the sequence without any conserved domain (variable region).
V1l	Start of a conserved domain (here V1, l = leftmost part).
=	Continuation of a conserved domain.
>	Indicates that one conserved domain goes into (overlaps) the next. Domains are normally not overlapping, so this could be an indication of a bad input sequence.

The line of asterisks indicates the end of one set of matches. Note that the graph should be viewed with a non-proportional font, such as Courier, if loaded into, e.g., Word.

Extraction results table

The full results of the Metaxa2 extraction are saved to a file with the suffix ".extraction.results". This file consists of tab-separated columns containing various data on each SSU sequence found. The file can be easily imported into spreadsheet programs such as Excel. The contents of the columns (from left to right) are explained in this table:

Column	Description
ID	The identifier of the query sequence.
Length	The length of the query sequence.
Origin	A one-letter abbreviation of the sequence origin. A = archaeal, B = bacterial, C = chloroplast, E = eukaryote, M = mitochondrial 16S, N = mitochondrial 12S.
Strand	A zero (0) if the SSU/LSU was found on the main strand, a one (1) if it was found on the complementary strand.
Domains	The number of conserved domains for the most likely origin that was found in the sequence.
Average E-value	The average E-value for these domains.
Average score	The average score for these domains.
Start	The starting position of the first domain.
End	The ending position of the last domain.
First domain	The domain that is located first on the sequence.
Last domain	The domain that is located last on the sequence.
Chimera	The word "Chimeric" if the sequence was marked as a potential chimera. Empty if not. Sequences will only be marked as chimeric if the --allow_reorder option is turned off. Note that this is *not* a robust measure against chimeras of all kinds.
Specific origin information	A collection of data of *all* possible origins for the given query. Each entry is a space-separated list, containing the origin type, the number of domains of that type, the average E-value, and the average score, e.g. "N: 4 8.2e-11 43.475"

Extraction results table

If table output is turned on, Metaxa2 will save statistics of every profile set that the sequence in question matches to in a file with the suffix ".hmmer.table". This file consists of tab-separated columns containing information on the rRNA sequence found. The contents of the columns (from left to right) are explained in this table:

Column	Description
ID	The identifier of the query sequence.
Length	The length of the query sequence.
List of hits	Each new column contains information of a profile match. Each column is organized as follows: "<starting position> - <ending position>: <name of matching profile> (<score>, <E-value>)".

As in the graphical output file, the table file is divided into sections. Each section represents one group of sequences, and begins with the line "X matches on main strand:", and ends with a line of asterisks.

Classification results table

If table output is turned on, Metaxa2 will save statistics of every BLAST match that the sequence in question produces against the database to a file with the suffix ".blast.table". This file consists of tab-separated columns containing information on the matches found, one

BLAST match per line. The contents of the columns (from left to right) are explained in this table:

Column	Description
Query ID	The identifier of the query sequence.
Subject ID	The identifier of the matching database sequence.
Score	The score this match has obtained in the classification system
Species	The species name of the database system
Score	The BLAST score of the match
E-value	The E-value of the match, as reported by BLAST

Each new query is indicated by a comment line, e.g.:

```
## Query AATT01000235.146421.147977|E
```

List of non-target sequences

If not-found output is turned on, Metaxa2 will write a list of sequences for which no conserved SSU/LSU regions could be found to a file with the suffix "_not_found.txt". The file contains only the identifiers of the non-rRNA sequences.

Sequence alignments

By default, Metaxa2 saves alignments of sequences of uncertain origin to a directory with the suffix "_alignments". The user may specify to instead align all SSU sequences by using the "--align all" option (note that this would increase the runtime (and the required disk space) significantly). The five best BLAST matches are aligned to the query sequence, and saved to an aligned FASTA file with the name "<query identifier>.aligned.fasta".

Chimeric sequences

If the option --allow_reorder is turned off, Metaxa2 will save an additional FASTA file containing sequences that are suspected to be chimeric. These are sequences with domains located in the wrong order. This is useful on full-length or near full-length data sets, but should not be used on short reads as it could increase the number of false negatives when run on short sequences.

Raw data

If the option to save all raw data is turned on, Metaxa2 will save all data from the pre-processing, HMMER-search, BLAST-search, as well as a file of raw statistics into a directory with the suffix "_metaxa_raw_output".

5. Metaxa2 Taxonomic Traversal Tool

In addition to the improved classifier, Metaxa2 also introduces a new bundled tool to further analyze the taxonomy output. This tool, called the Metaxa2 Taxonomic Traversal Tool – metaxa2_ttt, summarizes the taxonomic output of Metaxa2 at different taxonomic levels. Simply put, the traversal tool goes through the taxonomic predictions in the ".taxonomy.txt"

output file, and counts the number of entries associated with each taxonomic level. The levels are, roughly, corresponding to kingdoms, phyla, classes, orders, families, genera, species, and subspecies, in some cases followed by more specific annotations. The traversal tool can also filter the output according to reliability score, alignment length, percent identity to the best BLAST match and/or taxonomic group. The output of metaxa2_ttt consists of a number of tab-separated text files containing group counts at different taxonomic levels (by numbers), and a summary file with the suffix ".taxonomy.summary.txt". Each of the count files has the following format:

```
Bacteria;Actinobacteria;Actinobacteria      5
Bacteria;Firmicutes;Bacilli    14
Bacteria;Firmicutes;Clostridia        1
Bacteria;Proteobacteria;Betaproteobacteria        2
Bacteria;Unclassified Bacteria;       3
```

Each line here represents one node in the taxonomic tree, and the second column contains the number of entries associated with that node. The options for metaxa2_ttt are given in this table:

Options:

-i {file}	Metaxa2 taxonomy output file to process (with suffix ".taxonomy.txt") to use as input for metaxa2_ttt.
-o {file}	Base for the file names of the output files. Suffixes will be added automatically. Defaults to "metaxa_ttt_out".
-t {b, bacteria, a, archaea, e, eukaryota, m, mitochondrial, c, chloroplast, A, all, o, other}	Include only classifications of this taxonomic type(s). Several types can be specified, separated by comma. Default is to use all types (all).
-r {value}	Reliability cutoff. Entries with reliability scores below this cutoff will be classified as "unknown". Default = 0.
-l {value}	Length cutoff (in bp) for the best BLAST hit. Entries below this cutoff will be classified as "unknown". Default = 50
-d {value}	Percent identity cutoff for the best BLAST hit. Entries below this cutoff will be classified as "unknown". Default = 0.
-m {integer}	Maximum resolution level for taxonomic traversal. Providing a zero (0) to this option will remove this limit. Default is to have no upper limit for traversal (0).
-n {integer}	Minimum resolution level for taxonomic traversal. The kingdom level is defined as 1, and is the lowest possible setting. Default = 1.
-s {T or F}	If true, metaxa2_ttt will investigate only the last taxonomic level (which in the best case will represent a species resolution). Note that this setting will classify sequences at very different taxonomic levels! When true, metaxa2_ttt will only output the ".level_1.txt" output file. Default is off (F).
--remove_na {T or F}	If true, sequence entries with no BLAST hits will be set to 'Unknown'. Default is on (T).

Output options:

--summary {T or F}	Outputs a summary of results. On (T) by default.
--lists {T or F}	Outputs a list of counts for different taxa, one file for each traversal level. On (T) by default.
--separate {T or F}	Outputs the taxonomy file used as input, but separated for the different origins. Off (F) by default.
--unknown {T or F}	Outputs a list of entries designated as unknowns, with their statistics. Off (F) by default.

Information options:

-h	Displays the help message.
--help	Displays the help message.
--bugs	Displays the bug fixes and known bugs in this version of Metaxa.
--license	Displays licensing information.

The Metaxa2 Taxonomic Traversal Tool is designed to be run after Metaxa2 has completed, and uses the ".taxonomy.txt" output file as its input.

6. Metaxa2 Diversity Tools

Included with Metaxa2 since version 2.1 is also a set of other handy tools for taxonomic and diversity studies, collectively known as Metaxa2 Diversity Tools. These currently include four tools for automating and improving the analysis of the taxonomy output from Metaxa2. All these tools works with the *.taxonomy.txt output files of Metaxa2 or the output from the Metaxa2 Taxonomic Traversal Tool (metaxa2_ttt), and are intended to run after the normal Metaxa2 analysis has finished. The four tools are the Data Collector (metaxa2_dc), the Species Inference tool (metaxa2_si), the Rarefaction analysis tool (metaxa2_rf), and the Uniqueness of Community analyzer (metaxa2_uc). Each tool is briefly described with command-line options explained below.

Metaxa2 Data Collector (metaxa2_dc)

The Metaxa2 Data Collector is designed to merge the output of several *.level_X.txt files from the Metaxa2 Taxonomic Traversal Tool into one large abundance matrix, suitable for further analysis in, for example, R. Note that all arguments that are not preceded by a "flag" will be interpreted as input files for metaxa2_dc.

Options:

	All arguments not preceded with an option flag will be interpreted as input files for metaxa2_dc. Note that you can end the command-line by *level_6.txt to include all files ending with "level_6.txt" present in a particular directory.
-o {file}	The output file containing the full abundance matrix. Defaults to "collected_data.txt".
-t {integer}	Column containing the taxon name data. Default is zero (the first column).

-c {integer}	Column containing the count data. Default is one 1 (the second column).
-r {string}	String to be removed from each file name for use as the sample name. Regular expressions can be used. Default is '.level_[0-9].txt'.
-p {string}	Regular expression pattern for selecting the sample name from the file name. Default is '.*', which will cover full file name.

Information options:

-h	Displays the help message.
--help	Displays the help message.
--bugs	Displays the bug fixes and known bugs in this version of Metaxa.
--license	Displays licensing information.

Metaxa2 Species Inference Tool (metaxa2_si)

The Metaxa2 Species Inference tool can be used to further infer taxon information on, for example, the species level at a lower reliability than in the actual Metaxa2 classifier, using a complementary algorithm. The idea is that is if the only species present in, e.g., the Flavobacteriaceae family is *Ornithobacterium rhinotracheale* and a read is assigned to the Flavobacteriaceae family, but not to the species level, that sequence will be inferred to the *Ornithobacterium rhinotracheale* species given that it has more than 97% sequence identity to its best reference match. This can be useful if the user really needs species or genus classifications but many organisms in the studied species group have similar rRNA sequences, making it hard for the Metaxa2 classifier to classify sequences to the species level. The metaxa2_si tool works on the *.taxonomy.txt output from the Metaxa2 software.

Options:

-i {file}	Input taxonomy file from Metaxa2 (*.taxonomy.txt).
-o {file}	The output file containing the full abundance matrix. Defaults to the input name with the suffix ".inferred.txt" appended.
-l {integer}	Taxonomic level for performing inference (1 = domain, 2 = phylum, 3 = class, 4 = order, 5 = family, 6 = genus, 7 = species). Default is 7 (species level).
-c {value}	Percent identity cutoff to closest reference sequence for allowing inference. Default is 97.
--list_all {T or F}	Outputs a list of all possible taxa associations for sequence entries with multiple possible inferences. Off (F) by default.
--multiple {keep,merge,remove,assign}	Decides how to handle entries with multiple possible inferences. The 'keep' option will retain them in the taxonomy file without changes to their taxonomic classification, 'merge' will keep the entries but remove all taxonomic information and replace it with "Multi-origin sequence", 'remove' will exclude entries with multiple possibilities entirely, and 'assign' will randomly assign the entry to a possible taxa within the same taxonomic group. Default is to 'keep' entries (without modifications).
--low_identity {keep,merge,remove}	Decides how to handle entries with sequence identity below the specified cutoff (see the "-c" option). The 'keep' option will retain these entries in the taxonomy file without changes to their

taxonomic classification, 'merge' will keep the entries but remove all taxonomic information and replace it with "Low-identity sequence", and 'remove' will exclude entries with low sequence identity entirely, Default is to 'keep' these entries.

Information options:

-h	Displays the help message.
--help	Displays the help message.
--bugs	Displays the bug fixes and known bugs in this version of Metaxa.
--license	Displays licensing information.

Metaxa2 Rarefaction analysis tool (metaxa2_rf)

The Metaxa2 Rarefaction analysis tool performs a rarefaction analysis based on the output from the Metaxa2 classifier (or the Metaxa2 Species Inference tool). The tool produces three different curves, one for only the observed taxa (the *Observed number of taxa* column), one for the maximum number of taxa possible, given that all unknown entries are individual taxa (the *Theoretical maximum number of taxa* column), and one modeled column that takes the proportion of unknown entries into account and then models the number of different taxa sampled at a particular number of rRNA sequences drawn (the *Modeled number of taxa* column).

Options:

-i {file}	Input taxonomy file from Metaxa2 (*.taxonomy.txt).
-o {file}	Base for the name of output file(s). Defaults to "metaxa2_rf_out".

Entry selection options:

-t {b, bacteria, a, archaea, e, eukaryota, m, mitochondrial, c, chloroplast, A, all, o, other}	Include only classifications of this type(s). Several types can be provided, separated by commas. Default is 'all' types.
-r {value}	Specifies the reliability cutoff. Entries below the cutoff will be regarded as 'unknown' in the rarefaction analysis. Default is 0 (will include all entries).
-l {value}	Specifies the length cutoff (in bp) of the overlap to the best hit. Entries below the cutoff will be regarded as 'unknown' in the rarefaction analysis. Default is 50.
-d {value}	Sets the percent identity cutoff to the best hit. Entries below the cutoff will be regarded as 'unknown' in the rarefaction analysis. Default is 0 (will include all sequences, regardless of identity).
-m {integer}	Maximum resolution level for taxonomic traversal. Zero is unlimited. Default is not to limit this (0).
-n {integer}	Minimum resolution level for taxonomic traversal, starting at level 1. Default is 1 (domain level).
-s {T or F}	Investigate only the last taxonomic level (in good cases corresponding to species resolution). Default is not to do this (F).

-u {T or F}	If turned on, unclassified entries will be treated as unknowns. Default is off (F), which will use as much taxonomic information as possible.
--remove_na {T or F}	Set sequence entries with no BLAST hits to 'Unknown'. Default is to do (T).

Rarefaction options:

--resamples {integer}	Number of resamplings of the abundance data. Default is 1000.
--write {integer}	Write interval for the sampled list output. Default is 1 (which will write out every line).
--size {integer}	Fix the total number of sequences to this number. Allows for the addition of "artificial" entries to the analysis. By default this is set to the sum of all counts in the dataset.
--scale {integer}	Scale all samples to have this number of sequences. Default is 0, which turns scaling off.
--exclude_rows {list of integers}	A comma-separated list of rows to *not* to include in the analysis. Default is not to exclude any entries.

Output options:

--summary {T or F}	Turns on output of results summary. On (T) by default.
--lists {T or F}	Turns on output of lists of counts for different taxa, one file for each traversal level. On (T) by default.
--separate {T or F}	Will output rarefaction analysis results separately for the different origins. On (T) by default.
--unknown {T or F}	Outputs a list of entries designated as unknowns, along with their statistics. Off (F) by default.
--sampled {T or F}	Turns on output of a table containing the number of individuals sampled for different taxa at different number of sequences sampled. One file will be output for each traversal level. Note that this option can be extremely memory demanding, and that the resulting table can be very large. Off (F) by default.

Information options:

-h	Displays the help message.
--help	Displays the help message.
--bugs	Displays the bug fixes and known bugs in this version of Metaxa.
--license	Displays licensing information.

Metaxa2 Uniqueness of Community analyzer (metaxa2_uc)

The Metaxa2 Uniqueness of Community analyzer is a tool that allows analysis of whether the community composition of two or more samples or groups is significantly different. Using resampling of the community data, the null hypothesis that the taxonomic content of two communities is drawn from the same set of taxa (given certain abundances). The metaxa2_uc tools works on the output from metaxa2_dc or any abundance matrix with the same structure. The most important parameter to metaxa2_uc is the mode used for resampling. By

default, the comparison will be made to a distribution estimated from the average abundances in each group, with a certain degree of variation allowed. This model generates each resampled abundance number as follows:

$$[\text{Resampled abundance}] = [\text{True group relative abundance}] \pm [\text{Random number between 0 and 1}] * 2 * [\text{Standard deviation of group relative abundance}]$$

Options:

-i {file}	Input abundance matrix, e.g. derived from metaxa2_dc.
-o {file}	Base for the name of output file(s). Defaults to "metaxa2_uc_out".
-g {file,string,auto,none,all}	A file or string describing the sample group division, or 'auto' if the groups should be guessed from sample names, or 'none' if all samples should be treated individually, or 'all' if all samples should be treated as they come from the same group. Default is to use all samples as one single group ('all').
-r {value}	Number of resampling rounds for each sample. Default is 10000 (which will give a lowest p-value of 0.0001). Caution: memory usage *and* processing time largely scales with this parameter and it is recommended to lower it if you have hardware constraints.
-s {value}	Number of entries sampled in each resampling round for each sample. Higher number will improve sensitivity to differences. If this option is set to "min" the entries sample will be the number of entries in the smallest sample. Default is 1000.
-c {string}	Sample to compare to. Leave blank to compare to all samples. Use 'groups' to compare groups instead of samples. Default is blank, which will compare all samples to each other.
-w {value}	Within-sample variation cutoff in the range of 0 to 1 to compare to (corresponding to the proportion of all resampling values included). Default is 0.99.
-m {empirical,average,model}	The model to use for the resampling procedure. The 'empirical' mode uses the empirical distribution of abundances from each sample within each group, which will give rise to a large variance, which may be somewhat more realistic, but not statistical good-practice since it will estimate the parameters from the same data as it analyses. The 'average' mode assumes that the abundances of taxa in every sample of each group are drawn directly from the group average, which gives very small variance, and is thus unrealistic. The 'model' mode, finally, assumes a model that take within-group sample variation into account when drawing from the group average, which gives larger variance but in a more realistic way, and without (extensive) use of the data itself to estimate the parameters. Default is to use the 'model' (based on empirical testing of the software performance).
-d {bray,jaccard,euclidean}	Distance/dissimilarity measure to use when comparing samples and groups. The 'bray' option corresponds to the Bray-Curtis dissimilarity, 'jaccard' corresponds to the Jaccard distance, and "euclidean" corresponds to the Euclidean distance between samples/groups. Default is to use Bray-Curtis dissimilarity ('bray').
--binary {T or F}	Use presence/absence for distance/dissimilarity rather than taking abundances into account. Off (F) by default (will use abundance data).

--filter {value}	Filter out abundance values below this cutoff value. Default is 0, which will use all data.

Output options:

--summary {T or F}	Outputs a readable summary file of the results. On (T) by default.
--table {T or F}	Outputs a tab-separated table of the results. Off (F) by default.
--matrix {T or F}	Outputs the results in a matrix format. Off (F) by default.
--resampling_table {T or F}	Outputs the resampling table (which can be huge). Off (F) by default.

Information options:

-h	Displays the help message.
--help	Displays the help message.
--bugs	Displays the bug fixes and known bugs in this version of Metaxa.
--license	Displays licensing information.

An example workflow based on Metaxa2 and the diversity tools

The following section will outline a simple example workflow for how Metaxa2 can be used in taxonomic and diversity analysis. We will use the test file that comes with Metaxa2 to illustrate how data can be analyzed.

1) Running Metaxa2 on the input data:

```
metaxa2 -i test.fasta -o SSU_EXAMPLE --cpu 2
```

This should generate the following (excerpt from the SSU_EXAMPLE.summary.txt file):

```
Number of SSU rRNA sequences to be classified by Metaxa:     51
Number of SSU rRNA having at least one database match:       50
Number of SSU rRNA successfully classified by Metaxa:        50
Number of uncertain classifications of SSU rRNA sequences:   0
Total number of classifications made by Metaxa:              50
Number of SSU rRNA sequences assigned to each origin:
  Archaea:       10
  Bacteria:      10
  Eukaryota:     10
  Chloroplast:   10
  Mitochondria:  10
  Uncertain:     0
```

2) In the next step, we will generate some data to compare to. To keep it simple, we will compare to the LSU data from the same file:

```
metaxa2 -i test.fasta -o LSU_EXAMPLE --cpu 2 -g lsu
```

3) Now, we want to count the number of species, genera etc. present in these two datasets. For this we will use the Metaxa2 Taxonomic Traversal Tool (metaxa2_ttt). Note that we use the same base for the output names as for the input:
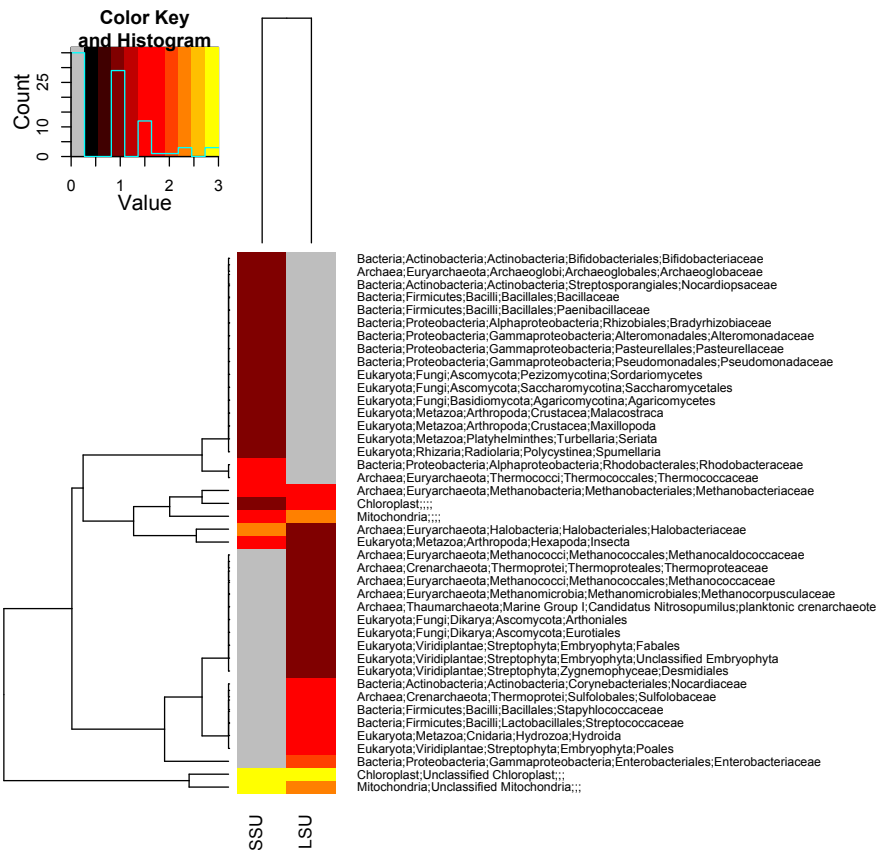
```
metaxa2_ttt -i SSU_EXAMPLE.taxonomy.txt -o SSU_EXAMPLE
metaxa2_ttt -i LSU_EXAMPLE.taxonomy.txt -o LSU_EXAMPLE
```

4) To be able to compare the LSU and SSU data we have generated, we want to combine the counts to a single abundance matrix. For this, we will use the Metaxa2 Data Collector (metaxa2_dc). Since the data is scarce in this example, we will combine the counts on the family level (level 5). To only get the first three characters of the file name as the header for each column, we use a regular expression for the pattern option (-p "^..."). Notice the neat trick we use to get both files using the wildcard (*) on the command line:

```
metaxa2_dc -o EXAMPLE_COMBINED.txt -p "^..." *level_5.txt
```

From this combined abundance matrix file, we can for example generate a heatmap of family abundances (this particular example was generated in R using the gplots library):

5) We can also use the abundance matrix to test if the SSU and LSU "communities" seem to be drawn from the same species pool, that is if they significantly differ. For this we will use the Metaxa2 Uniqueness of Community Analyzer (metaxa2_uc). We will use the "auto" mode to detect the groups, which will separate the LSU and SSU datasets into separate groups. Note that we also turn on the matrix and table output, to be able to analyze and visualize the results more efficiently later. Note also that this tool takes a rather long while to run, even on this small dataset:

```
metaxa2_uc -i EXAMPLE_COMBINED.txt -o EXAMPLE_COMBINED --table T --
matrix T -g auto
```

This analysis will generate the following output, indicating that if we assume that the LSU sequences are drawn from the SSU sequence pool and vice versa, the datasets significantly differ. However, if we assumed that both datasets are drawn from a single, common, family pool, the datasets do not differ significantly to that total pool (as seen when compared to the "_all_" group):

```
Results:
==================================================
Sample Group Bray-Curtis dissimilarity (min-max) p-value Significance
--------------------------------------------------
LSU     LSU     0.347   (0.347-0.348)   < 9.996e-01
LSU     SSU     0.698   (0.685-0.722)   < 9.999e-05     ***
LSU     _all_   0.301   (0.248-0.363)   < 9.984e-01
--------------------------------------------------
Internal distance (LSU, 0.99):  0.347
==================================================


Sample Group Bray-Curtis dissimilarity (min-max) p-value Significance
--------------------------------------------------
SSU     LSU     0.657   (0.636-0.676)   < 9.999e-05     ***
SSU     SSU     0.348   (0.348-0.350)   < 9.988e-01
SSU     _all_   0.296   (0.239-0.353)   < 9.997e-01
--------------------------------------------------
Internal distance (SSU, 0.99):  0.348
==================================================
```
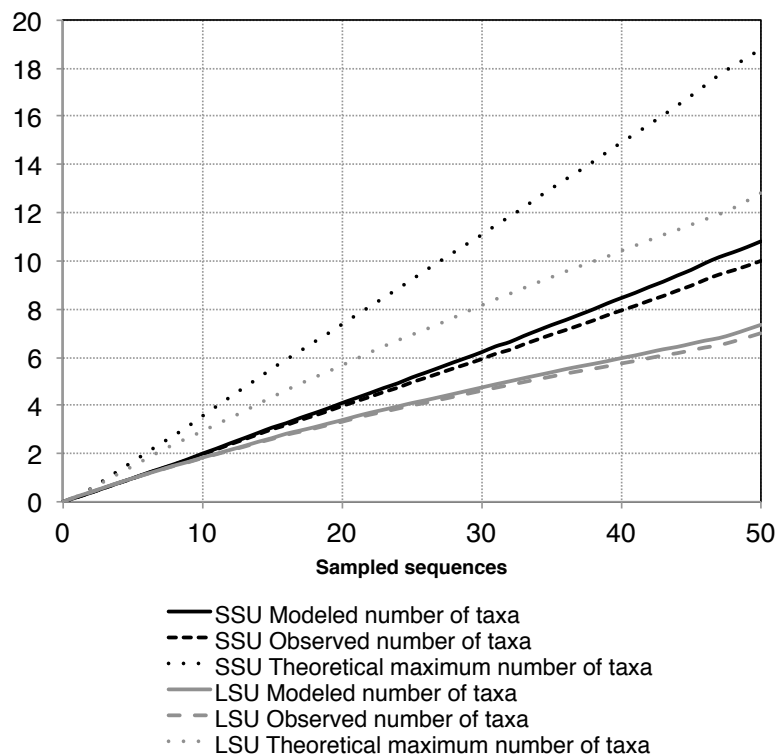
6) As a next step, we might want to try to further infer species classifications using the Metaxa2 Species Inference tool (metaxa2_si). This tool takes the original taxonomy output from Metaxa2 as input, so we will need to return to these files:

```
metaxa2_si -i SSU_EXAMPLE.taxonomy.txt -o SSU_EXAMPLE_INFERRED.txt
metaxa2_si -i LSU_EXAMPLE.taxonomy.txt -o LSU_EXAMPLE_INFERRED.txt
```
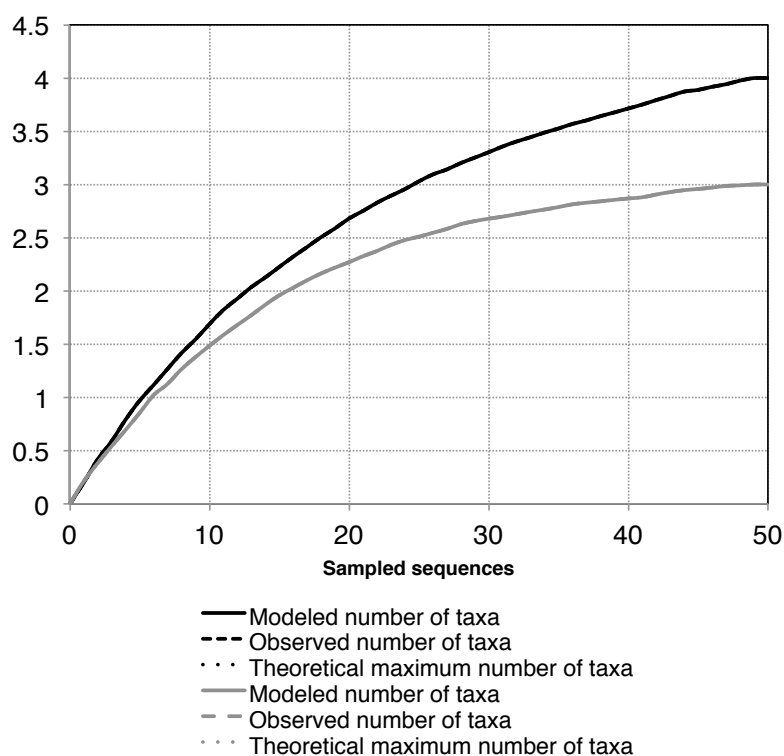
7) Using the inferred species data, we may now attempt to do a rarefaction analysis for each taxonomic level, using the Metaxa2 Rarefaction tool (metaxa2_rf). This tool uses the taxonomy output from the Metaxa2 classifier *or* inferred taxonomy data from the Metaxa2 Species Inference tool as input:

```
metaxa2_rf -i SSU_EXAMPLE_INFERRED.txt -o SSU_EXAMPLE_RF
metaxa2_rf -i LSU_EXAMPLE_INFERRED.txt -o LSU_EXAMPLE_RF
```

This will result in a number of files (one for each organism group and each taxonomy level). By loading the tab-separated SSU_EXAMPLE_RF.bacteria.rf.7.txt file in, e.g., Excel, it is possible to plot rarefaction curves. In this example we have combined it with the rarefaction curves for the LSU data (in grey):

As you can clearly see, this data is not well saturated, so we can also repeat the analysis on class level (level 3) for the purpose of illustration:

At this level, the modeled and observed curves completely overlap (because all classes in the data has been classified, so there are no unknown entries), and the curve for the LSU seem to start to saturate at around 45, while the SSU curve (black) does not look completely saturated.

It is of course possible to do much more with these tools, but this is an example of how the tools included with Metaxa2 can be used to make simple analyses of community composition and diversity.

7. Internal changes in Metaxa2

The original algorithms and design behind Metaxa are described in the manual for Metaxa version 1.1.2 (see part 4: *Algorithm and implementation* in that manual). This manual will mainly focus on the changes made from previous Metaxa versions, instead of reiterating the information from the previous manual.

If the main design goal for the original version of Metaxa was to achieve fast and accurate extraction of SSU sequences in large data sets, without introducing a large number of false positives, the design goal of Metaxa2 is to do the same task faster, better and for larger data sets. In addition, Metaxa2 adds the ability to extract and classify LSU sequences, as well as to produce more precise taxonomic classifications. Metaxa2 still relies on the HMMER3 software, which allows for extremely fast comparisons filtration of the input dataset, and further analysis of just a subset of the input data. Similarly to version 1, Metaxa2 uses multiple HMM profiles representing conserved domains in the SSU sequence. In addition, separate sets of HMM profiles for SSU sequences of archaeal, bacterial, eukaryal, mitochondrial and chloroplast origin are utilized. We have also added HMM profiles representing the LSU gene in these lineages, in very much the same way. To avoid false positive matches, Metaxa2 by

default requires at least two such conserved domains to be found in a query sequence. This criterion brings down the false positive rate to close to zero.

The subsequent BLAST-based classification step now has an even more elaborately crafted and curated database, allowing Metaxa2 to make accurate taxonomic predictions that often go down to genus or species levels. The scoring system for assigning the sequences to archaeal, bacterial, eukaryal, mitochondrial or chloroplast origin still remains in Metaxa2. If the origin of the final classification does not agree with the predicted origin from the HMMER-based step, the sequence classification is marked as uncertain (by applying a "#" to the end of the definition line). The sequence is also marked as uncertain if the difference between the scores of the two most likely origins is smaller than the number of sequences of analyzed BLAST-matches (by default 5).

Most changes that have been introduced in Metaxa2 are related to either the improved taxonomy engine, or adaptions to modern metagenomics, which is moving towards even larger datasets, but with short read lengths. In general, the internal changes are invisible to the end user, but some of them might have impact on particular usage scenarios.

First of all, the main metaxa2 program now is able to perform quality filtering of sequences in the FASTQ format (http://en.wikipedia.org/wiki/Fastq). Please note that the filtering is performed before metaxa2_x or metaxa2_c has been started. Thus, none of the individual programs can read FASTQ-format. Instead, they still expect input in FASTA format.

Metaxa2 accommodates for paired-end reads in a special fashion. Similarly to FASTQ input, paired-end libraries must be pre-processed through the first metaxa2 program and cannot be directly inputted into metaxa2_x or metaxa2_c. This is because Metaxa2 re-organizes the input sequences into a special concatenated format, which enables HMMER to work on both ends *at once*. This conversion, however, must happen before the extraction step can begin. This design decision was taken in order to streamline the downstream processing of sequence entries. Although it would be possible to re-write the code so that the individual tools handle FASTQ and paired-end input, it would severely complicate the way Metaxa2 internally manages its files. When Metaxa2 reads in a paired-end library, it first concatenates the two reads into one sequence, retaining information on where the original reads begin and end. The paired-end information is saved to a file called "pairinfo.1.txt", which can be retained and accessed by running Metaxa2 with the "--save_raw T" option. The metaxa2_x program reads the concatenated FASTA file and the pair data to create an output file containing the identified rRNA sequences. In this file, Metaxa2 also inserts an insert sequence consisting of repeated "N" characters between the reads in the pair. This file is then used as a regular FASTA file as input to metaxa2_c. Thus, metaxa2_c does not consider the paired reads; rather it looks at each concatenated entry as one single, long FASTA sequence with a stretch of unknown nucleotides in it.

The Metaxa2 Extractor (metaxa2_x) has two major changes to it since the last version. First of all, it is now "gene-agnostic", that is it is not hardcoded for SSU genes anymore. In theory, that means that any marker gene or region could now be used with Metaxa2; in practice, however, it only means added support for the LSU rRNA gene. This can be seen in the naming of the HMM profiles, which now are ordered sequentially without notions of e.g. "V1l" and "V3r". Secondly, the extractor has a different way of handling the input sequences; partially inspired by the work we did on ITSx (http://microbiology.se/software/itsx/). In principle, this means that Metaxa2 does not process sequences sequentially. Instead, batches of sequences (the number dependent on the available system memory) are kept in RAM, and processed separately. If you choose to use the "--save_raw T" option, you may notice that there are files containing ".1." and higher numbers among the raw data files. Those are the batches processed by Metaxa2. This way of handling sequences greatly speeds up the Metaxa2 post-processing of HMMER output (which becomes a time-limiting step in typical Illumina datasets). Minor such changes were introduced already in Metaxa 1.1, but have now been fully implemented to accommodate for increasingly larger datasets. The sequence

handling has also gone through some minor changes to be able to deal with paired-end sequences. However, this should not affect the behavior on single-end sequences.

The classifier (metaxa2_c) has undergone a complete overhaul when it comes to making taxonomic predictions. The new classification system is built upon the taxonomic information of the (by default) five best BLAST matches to each rRNA in the input data (this can be changed using the "-M" option). For each rRNA entry, Metaxa2 compares the taxonomic affiliation of the top BLAST match with the second one and so on. In each comparison the percent identity to the query is taken into account. If the BLAST matches point to the same taxonomic origin, the query sequence gets a taxonomic affiliation with a high reliability score (close to 100). If not (that is if the score, by default, is below 80 (specified by the "-R" option)), the comparison is repeated at the taxonomic level above (e.g. genus if the last comparison was made on species level), until the score is above 80. In this way, all sequences get a taxonomic affiliation at a trustworthy taxonomic level. This data is written to a file with the suffix ".taxonomy.txt".

The default settings of Metaxa2 should be useful in most situations. However, since the software has been tweaked to accommodate for analysis of really large datasets, you should consider if they are suitable for your purposes and for your data set. If the data set is small, the user should consider running the software multiple times on the data, with different settings, and analyze the outcome. On larger data sets, it might be more feasible to only run Metaxa2 on a subset of the sequences for testing. The graphical output is very useful for determining whether Metaxa2 performs as desired on long-read (>400 bp) data, as the positions of the found conserved domains can be examined easily. If domains are missing, the criteria might be set to be too stringent. If they are not in sequential order (from V1l to V9r for SSU sequences), that might be an indication that there is something wrong with the input sequences.

The HMMER program hmmsearch, used by Metaxa2, uses heuristic filters to increase the search speed. Metaxa2 runs hmmsearch as it is, with the "--max" option disabled. To turn off all heuristic filters of HMMER, Metaxa2 can be run with the "--heuristics F" option. This increases detection power at the cost of speed.

8. Running the analysis steps of Metaxa2 separately

The analysis procedure of Metaxa2 is divided into three steps: pre-processing, extraction and classification. These steps are normally run in sequence through the metaxa2 command. However, they can also be run separately if the user wishes. To run the extraction step independently, use the metaxa2_x command. This command takes a subset of the Metaxa2 options (other options will be ignored). To see the available options for the metaxa2_x command, type "metaxa2_x --help" on the command line. To run the classification step on a set of known rRNA sequences, use the command metaxa2_c. The options for metaxa2_c can be seen by typing "metaxa2_c --help" on the command line. Note that the output files obtained when running each step separately will be slightly different than those obtained through running the entire Metaxa2 pipeline. Additionally, FASTQ and paired-end file processing is ***not*** supported when running the analysis steps separately.

9. 'Undocumented' features

Metaxa2 has three undocumented options that can be activated, but they are considered *experimental* and should be used with caution. One allows you to use Usearch instead of the normal BLAST implementation for the classification step of Metaxa2, another allows using a set of additional HMM profiles for the SSU extraction, and the final allows seeing the underlying taxonomic data justifying the taxonomic predictions. Note that the Usearch

support is considered experimental and it is not recommended that this option is used in sharp analysis.

'Undocumented' options:

-t {o, other}	It is possible to supply an additional set of HMM profiles in an O.hmm file within the HMMs directory. This custom set can be any type of profiles, but the profiles must be named according to the convention in the other HMM files.
--taxonomy {complete}	It is possible to add the underlying taxonomic data that the taxonomic predictions are based on by using the "--taxonomy complete" option instead of setting it to "--taxonomy T". This data is added to each entry in the ".taxonomy.txt" output file.
--usearch {version}	Runs Usearch instead of BLAST for the classification step. Note that you need to specify the Usearch version used, as the algorthms and options differ. Off (0) by default.
--usearch_bin {path}	Specifies where Metaxa2 should expect to find the Usearch binary (either the name of the binary or the complete path to it) to be used. Default is 'usearch'.
--ublast {T or F}	Runs the Ublast algorithm instead of the normal Usearch algorithm. Default is to do this, since sensitivity is empirically better (T).

10. License information

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program, in a file called 'license.txt'. If not, see: http://www.gnu.org/licenses/.

Copyright (C) 2011-2016 Johan Bengtsson-Palme et al.