User's guide: Manual for Mumame

This is a guide on how to install and use the software utility Mumame, version 1.0. The software is written for Unix-like platforms, and should work on nearly all Linux-based systems, as well as macOS.

Contents of this manual

1. Detailed installation instructions
2. Usage and commands
    a. mumame
    b. mumame_build
    c. analyze_mumame_output.R
3. Input data for the database builder
4. Statistical interpretation
5. License information

1. Detailed installation instructions

First of all, Perl needs to be installed on the computer. Most Unix-based systems including Linux and macOS 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.

This version of Mumame relies on the USEARCH software, version 7 or later, which has to be installed on the computer for the software to operate. USEARCH can be downloaded from http://drive5.com/usearch/. Use the software download link and download the software to a preferred location. Change the permission of the binary with a name starting with "usearch" using the following command: "chmod +x /usr/bin/usearch........". Move the USEARCH binary into your preferred bin directory and rename it "usearch" (otherwise Mumame will not be able to identify it). Some additional installation instructions can be found here: http://drive5.com/usearch/manual/install.html.

Go to http://microbiology.se/software/mumame in order to download the Mumame package. Download it to your preferred directory. Unpack the downloaded tarball with "tar -xvfz mumame_1.0b.tar.gz". A directory called Mumame will be created. You will see a number of files and directories inside it, including mumame, mumame_build, an R script and this user's guide. Enter the directory, and copy the mumame and mumame_build files to your preferred location for executable files. If successful, you should now see Mumame's help message when typing the command "mumame --help". If this does not work, try logging out and then in again and retry.

In order to install certain packages, you might need to have superuser privileges. If you don't have superuser privileges on your machine: Create a directory within your user directory, e.g. /home/user/bin/, and 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. Note, though, that 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.

2. Usage and commands

mumame

Mumame accepts input in the FASTA and FASTQ formats, including gzipped and DSRC compressed files. To list all the available options for Mumame, type "mumame --help". In the simplest case, Mumame is run by the command "mumame -d database -i input_file -o output". Several input files can be specified for the same run. Options without a "flag" prefix will be interpreted as input files.

Main options:

-i {file}	Input file in FASTA, FASTQ, GZIP or DSRC format.
-o {file}	The base name of the output files. Default is 'mapping_results'.
-d {mumame database}	The path to a database built using mumame_build. Note that the name should not contain e.g. the ".fasta" suffix of the database files.
-L	Turns off auto-detection of which input files belong to the same sample (e.g. in the case of paired-end input).
-n	Specifies that the database is in nucleotide format. Default is to assume protein sequences in the database.
-m {usearch}	Software to use for the read mapping. In this version, only "usearch" can be selected. Default is 'usearch'.
--alnout	Will save alignments to a separate file, if possible in the mapping software. Default is not to save the alignments.
--binary {path}	The full path to the software binary to use for the mapping. Not used by default.

Sequence selection options:

-c {ratio}	Sets the sequence identity cutoff for matches. Default is 0.95 (95% identity).
-t {ratio}	Sets the required target coverage for matches. Default is 0.55 (55% of the database sequence). Warning: Setting this number to 0.50 or below will likely cause false-positive detections!).
-q {ratio}	Sets the required query coverage for matches. Default is 0.

Information options:

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

mumame_build

The mumame_build tool is used to build databases for the Mumame software. It takes two input files: a set of sequecnes in FASTA format and a list of mutations to include in the database. The format for this mutation file is modelled after the ARO format introduced in the CARD database (https://card.mcmaster.ca). This means that you can supply the "snps.txt" file from CARD together with the appropriate FASTA file directly to mumame_build to construct a database containing all point mutations conferring antibiotic resistance from CARD (although this will take time to build). This data can be downloaded from here: https://card.mcmaster.ca/download. Look for the "Download CARD Data" files.

Main options:

-i {sequence file} Input file in FASTA format.

-m {mutation file The path to the mutation information input file (tab-separated text). See below for the format specification.

-o {file} The base name of the output files. Default is 'mutation_database'.

Database parameters:

-c {integer} Sets the size of the cut-out that mumame_build makes around each mutation position. Default is 20 for protein databases and 55 for nucleotide. This option overrides the "-r" option below.

-r {integer} Sets the expected minimum read length for the database. This will indirectly set the size of the cut-out that mumame_build makes around each mutation position. Not used by default. This option overrides the "-c" option above.

-n Specifies that the input for the database is nucleotide sequences. Default is to assume protein sequence input.

-t {fasta} Type of database to build. Currently, only "fasta" is supported. Default is 'fasta'.

-d {integer} Maximum depth to transcend into when building all combinations of mutations. Increasing this value increases the memory requirements exponentially. Default is 12.

Information options:

-h Displays the help message.

--help Displays the help message.

--bugs Displays the bug fixes and any known bugs in this version of Metaxa.

--license Displays licensing information.

analyze_mumame_data.R

The analyze_mumame_data.R is an R script that can be run through the R statistical software. The recommended way to run it is as follows:

1) Place the script in the same directory as where you keep your output data from a Mumame run (the .table.txt and .per-mutation.txt files).
2) Open the script in a text editor, such as TextEdit on macOS or NotePad on Windows. Do **not** change anything below the header of the script. If you do, we will not provide support in understanding weird output.
3) Edit the header section of the script according to the instructions given in the script itself. The instructions will tell you to enter the name of the input file and details of the experimental design.
4) Launch R.

5) Change the working directory to where the script and the output data are located.
6) Run the script by typing:

```
source("analyze_mumame_data.R")
```

The script may produce some warnings (but should produce no errors) and will write an output table (.mumame_stats.txt) and a PDF with plots (.plots.pdf) if successful.

Script options (entered in the header of the script prior to running it through R):

filename = Enter the name of the input file here. Usually it ends either with ".table.txt" or ".per-mutation.txt".

outputname = Enter the desired base for the output file names.

testVariable = This variable corresponds to the experimental design. It should be a comma separated list of values, encapsulated with c(......). The format for the experimental design is further outline below.

testVariableName = A description of what is tested for in the experimental design.

The experimental design could be constructed in two different ways. The first way is that it can correspond to a categorical variable, such as "exposed" vs. "control". In this case, control samples should be represented by zeros (0) and exposure or test samples should be represented by ones (1). In this version of Mumame, this script does not support several categories.

The second way is to specify a numerical testing variable, corresponding, e.g., to an exposure concentration. In this case, every item in the list should be a number. If a log transformation is desired for this list, it can be encapsulated with "log(c(...))" instead of just by "c(...)".

Make sure that the order of the items in the testVariable list corresponds *exactly* to the order of the samples (columns) in the Mumame output files. Note that each sample (column) in the Mumame output appears twice, first counting the mutation counts and then the wildtype count. For this script, however, each sample should only be specified once, but in the same order as they appear in the Mumame output file. For more information on the statistical interpretation, see the separate section on this further below.

3. Input data for the database builder

The mumame_build command takes as input a FASTA file and a table containing information on resistance mutations. The format of the second file is specified to be compatible with the "snps.txt" file provided by the CARD database for antibiotic resistance mutations. This file should contain the following columns, which could be placed in either order but are **required** to have these exact column names on the first row.

Mutation information table format:

Accession This column *typically* contains a sequence identifier corresponding to an entry in the input sequence file. However, to ensure compatibility with the CARD format, this can also be an ARO identifier. In the ARO case, the identifier needs to be present in the sequence header preceded by "ARO:" and ending with the "|" character. (This is how the CARD sequence files are distributed, so no modification is required if those are used.)

Name

This column contains a name for each mutation. Most often, this is a gene name, sometimes followed by a specific description of the phenotype the mutation causes, e.g. “conferring resistance to antibiotic X”. It is generally a good idea to also include the species where the mutation is identified in this column.

Model Type

This column can be either of “protein variant model” or “rRNA gene variant model”. However, in the current version of mumame_build this information is not used for anything, and column can be left out with full functionality.

Parameter Type

This column is most often specified as “single resistance variant” for point mutations, or occasionally “multiple resistance variants”. However, in the current version of mumame_build this information is not used for anything, and column can be left out with full functionality.

Mutations

This is the most important column, as it specifies which mutations Mumame should look for. Each mutation should be given in the format of “G81C”, where the first character is the wildtype residue, the last character is the mutated residue and the number corresponds to the position in the sequence. Several mutations can be combined using commas, e.g. “A244T,G288W,V303G”.

Below, a few lines from the CARD “snps.txt” file are given as an example of what the file can look like:

Accession	Name	Model Type	Parameter Type	Mutations
3003817	Acinetobacter baumannii gyrA conferring resistance to fluoroquinolones	protein variant model	single resistance variant	G81C
3003817	Acinetobacter baumannii gyrA conferring resistance to fluoroquinolones	protein variant model	single resistance variant	S83L
3003818	Acinetobacter baumannii parC conferring resistance to fluoroquinolone	protein variant model	single resistance variant	D87E
3004049	Escherichia coli fabG mutations conferring resistance to triclosan	protein variant model	single resistance variant	Y151V
3003326	Mycobacterium tuberculosis embB with mutation conferring resistance to ethambutol	protein variant model	single resistance variant	R507G
3003326	Mycobacterium tuberculosis embB with mutation conferring resistance to ethambutol	protein variant model	multiple resistance variants	A314G,Y322C

4. Statistical interpretation

The R script included with Mumame will output a table of effect sizes and significances for the detected mutations. Note that the p-values in this table are **not** corrected for multiple testing. The table has five columns, and is structured like this:

mumame_stats.txt format:

Mutation

This column includes information on which mutation and description the tests describe.

Ef-mut+test

This column contains the effect sizes for the interaction between the testing and mutation variables for each tested mutation in an overdispersed Poisson generalized linear model accounting for differences in sequencing depth. The magnitude of this effect size is very hard to translate into a biological interpretation, so therefore it is recommended that only the sign of the effect is considered. A positive value means that the test variable increases the mutation frequencies and a negative effect size means that the test variable decreases the mutation frequencies.

Ef-ratio

This column contains the effect sizes for the different proportions of mutated sequences between test and control samples. This effect size can roughly be interpreted as the difference in average proportions, but is weighted by sequencing depth.

p.anova

This column contains the p-values for each tested mutation in the ANOVA test based on comparing two overdispersed Poisson generalized linear models accounting for differences in sequencing depth. One of these models includes the testing variable as an explanatory factor, and the p-value describes the probability that this model describes the data better than a model not including the testing variable to explain the data.

p.ratio

This column contains the p-values for the different proportions of mutated sequences between test and control samples. The p-values describe the probability that the proportions between tst and control samples would differ as much as they do just by chance.

For every mutation detected the Mumame analysis script performs two different statistical tests. One is fitting of an overdispersed Poisson generalized linear model (GLM). This model is compared to a null model in which the testing variable is not included. Those two models are the compared using an ANOVA analysis. The other test is a comparison of the proportions of mutated sequences identified in the test and control samples. This comparison is also performed using a GLM, but assuming normally distributed data. This analysis is also weighted by the sequencing depth in the samples. Note that regardless of if a categorical or numerical testing variable is specified in the R script, the same tests are run on the data.

Which test is preferable depends on the input data. The Poisson model may have better detection ability when overall counts are low. However, it also seems prone to produce suspected false positives in our testing, calling for some caution when interpreting results from this model. The asterisks denoting significance in the generated plots are based on the ratio tests.

5. License information