User's guide: Manual for Megraft 1.0.1

This is a guide on how to install and use the software utility Megraft. The software is written for operating systems of the Unix family, including Linux, BSD, and 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 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 (or whatever shell configuration file (.bash_profile, .bashrc, …) employed by your shell) 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) - please note that Megraft will not work with earlier versions of HMMER. 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 must 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. When you have downloaded and unpacked the tarball, you will find the binaries in the binaries directory inside 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 NCBI-BLAST package (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/LATEST/) for sequence similarity searches. The current version of Megraft relies on BLAST, ***not*** BLAST+, and was written with version 2.2.24 in mind; it should, however, work with any 2.2.x version of BLAST. 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 output from BLAST.

Download and install MAFFT (http://mafft.cbrc.jp/alignment/software/) for multiple alignment. The current version of Megraft relies on MAFFT version 6. Instructions for installing MAFFT are available on the MAFFT download page.

Download GramCluster (http://bioinfo.unl.edu/gramcluster.php) and unzip the archive. Before following the instructions in the README file, find the file called "FileIO.c" in the "src" directory. Open it (this should be possible in any editor for plain text – make sure to save the file as a text file, not as e.g. a Word file), and change the line that reads:

```
#define FILEIO_MAXIMUM_HEADER_LENGTH_FOR_CLUSTERS      19
```

into a line that reads

```
#define FILEIO_MAXIMUM_HEADER_LENGTH_FOR_CLUSTERS       300     // ***
Modified threshold ***
```

You should then follow the instructions in the README file to install GramCluster. In short, you move into the “src” directory in the terminal and type “make clean” followed by “make”. Then copy the new file called “GramCluster” into the bin directory of your choice.

Go to http://microbiology.se/software/megraft in order to download the Megraft package. Download it to your preferred directory. Unpack the downloaded tarball with “tar xvfz megraft.tar.gz”. A directory called Megraft will be created. You will see the following files and directories inside it: megraft, the megraft_db directory (containing the Hidden Markov Models and a BLAST database), the user’s guide, the README.txt file, the license.txt file, and a test input file. Enter the directory and copy (or move) the file “megraft” and the directory “megraft_db” to your preferred bin directory. If Megraft is successfully installed you should see its help message when executing the command “megraft --help”.

2. Usage and commands

For the *very* impatient only: follow the brief installation instructions in the file README.txt. To try to graft the SSU fragments in the file test.fasta onto the reference sequences, you would type "./megraft -i test.fasta -o test" on the command line. Note that Megraft by default expects to find its database directory in the same directory as the megraft program itself.

For all other users: Megraft expects input in the FASTA format. It is possible to input both aligned and unaligned FASTA files, containing both DNA and RNA sequences. By default, Megraft outputs three files; one variability summary file of the entire run, one file with details on the input sequences, and a FASTA file of all extended SSU sequences. To list all the available options for Megraft, type “megraft --help”. You can use the test.fasta file that comes bundled with the software for a test run. This file contains 16 SSU fragments,

randomly generated to simulate typical 454 sequencing read lengths in a metagenomics effort. In the simplest case, Megraft is run by "megraft -i input_file -o output". Below is a listing of all options Megraft accepts.

Main options:

-i {file}	Nucleotide FASTA input file to process (i.e., the query file), with or without gap characters. If no input is specified, Megraft will abort.
-o {file}	FASTA output file to write expanded sequences to. Defaults to megraft_output.fasta.
--hmm {HMM-file}	A path to a file of HMM-profiles representing SSU rRNA conserved regions. By default, Megraft assumes to find the databases in the megraft_db directory, located in the same directory as Megraft itself, such that the user does not have to specify this parameter by default.
-d {database}	The BLAST database or FASTA-file used as reference for expansion. By default, Megraft assumes to find the databases in the megraft_db directory, located in the same directory as Megraft itself, such that the user does not have to specify this parameter by default.

Global options:

--domain {b, bacteria, a, archaea, e, eukaryota, 16S, 18S}	Set of conserved domain profiles to use for the search. Default is to use bacterial SSU profiles (the "b" option).
--mode {proxy, insert, full} (also -m {proxy, insert, full})	Selects the mode which Megraft will be running in, i.e. how it will create expanded sequences. Default mode is "full". You can read more about Megraft's modes in the "Algorithm and Implementation" section.
--cpu {value}	The number of CPU threads to use. Megraft performs significantly faster using more CPUs/cores. Default is 1.
--distance {value}	Controls the GramCluster grammar distance used in pre-clustering. Default is 0.05.
--no_clustering	Turns off the pre-clustering completely. Removes the dependency on GramCluster, but may produce less refined results.
--megablast	Uses megablast instead of regular BLAST for higher speed but lower accuracy.

Full model parameters:

--c_rate {value}	Uses a specific fixed mutation rate when introducing variation into the conserved domains of the output sequences. Default is not to do so.
--v_rate {value}	Uses a specific fixed mutation rate when introducing variation into the variable domains of the output sequences. Default is not to do so.
--c_corr {value}	Specify the correction factor for estimating variability in conserved domains. Smaller values than 1 reduce observed diversity, while higher values increase it. Default correction factor is 1, which does not change the observed variability.
--v_corr {value}	Specify the correction factor for estimating variability in variable regions. Smaller values than 1 reduce observed diversity, while higher values increase it. Default correction factor is 1, which does not change the observed variability.

--i_cut {value}	Sets the shortest grammar distance for insertion of an input fragment into the reference sequence. Default is 0.00.
--m_cut {value}	Sets the shortest grammar distance for introducing additional variation into the reference sequence. Default is 0.02.
--conserved_model {comma-separated list of values}	The model used for differential mutation rates of the conserved regions of the SSU sequence to indicate that the conserved regions still differ somewhat in their degree of conservation. The model consists of a list of numbers, one for each conserved region (including one for the beginning and end of the SSU sequence), totalling ten numbers for bacteria. The numbers must be separated by commas. By default, the choice of variability model is determined by the --domain option (explained above) such that there are three sets of values: bacteria, archaea, and eukaryote. For more information on the variability model see the "Algorithm and Implementation" section.
--variable_model {comma-separated list of values}	The model used for differential mutation rates of the variable regions of the SSU sequence to indicate that the variable regions differ substantially in their degree of conservation. The model consists of a list of numbers, one for each variable region (including an unused number for the the beginning of the SSU sequence), totalling ten numbers for bacteria. The numbers must be separated by commas. By default, the choice of variability model is determined by the --domain option, (explained above) such that there are three sets of values: bacteria, archaea, and eukaryote. For more information on the variability model see the "Algorithm and Implementation" section.

Information options:

-h	Displays the help message.
--help	Displays the help message.
--bugs	Displays the bug fixes and known bugs in this version of Megraft.

3. Output files

Megraft outputs three files. The default output consists of three items: a FASTA-file of expanded sequences, a file containing information on each input and reference sequence pair, and a variability summary file.

FASTA-output

Megraft generates a FASTA file containing sequences expanded by the program. The sequences in these files are marked according to what modifications they have undergone. If Megraft simply replaces the input sequence with a sequence from the reference database, the definition line will read "QUERY_ID replaced by REFERENCE_ID". If Megraft inserts the input fragment into a reference sequence, the line will say "QUERY_ID inserted into REFERENCE_ID", while if the full model has been used to introduce variability, the definition line will be "QUERY_ID expanded using the full method and sequence REFERENCE_ID". The output file may look like this:

```
>D060contigs1493 replaced by AY986069.1.1354
GATGAACGCTGGCGGCGTGCTTAACACATGCAAGTCGAACGAAGCACTTTATT…
>D059contigs1494 inserted into GQ898843.1.1499
GCCCTTAGTTCGATCCTGGCTCAGGATGAACGCTAGCTACAGGCTTAACACAT…
>D059contigs1438 expanded using the full model and GQ898843.1.1499
GCCCTTAGTTCGATCCTGGCTCAGGATGAACGCTAGCTACAGGCTTAACACAT…
```

Variability summary

A summary of the overall variability within each Megraft run is written to a file with the suffix ".variability.txt". This file contains two lines, the first representing the total variability between the conserved regions of the input and reference sequences, and the second representing the variability of the variable regions. These numbers provides a rough estimate of how well the sequences in the reference database represents the input data. Typically, the variability in conserved regions is lower than the variability in variable regions. Megraft generally produces better results when variability (and the standard deviation) is low. When variability values exceed 0.03, the user should start to be cautious about making diversity estimates based on the Megraft output. An example of a summary file is shown below:

```
Average variability in conserved domains:    0.000547       (Std. dev. 0.00157)
Average variability in variable domains:     0.000889       (Std. dev. 0.00204)
```

Sequence-to-reference comparisons

Megraft also writes information on how each individual input sequence differ from the closest reference sequence into a file with the suffix ".sequence_info.txt". For each input sequence, seven lines are written. The first represents the sequence header line, the second contains the ID of the best BLAST match in the reference database, line three and four shows the true variability between the input and reference sequence, and the last line shows the length of the output sequence. Line five and six contain the modified variability measure used by the full model and is printed just for reference. This file could look like this:

```
Sequence ID: >gi|254972987|gb|GQ329607.1| Uncultured Pseudomonas
Best BLAST match: JF178460.1.1358
Variability in conserved domains: 0.00409836065573771 (over 244 bp)
Variability in variable regions: 0 (over 107 bp)
Modified variability in conserved domains (for full model):
0.00327868852459016 (over 244 bp)
Modified variability in variable regions (for full model): 0 (over
107 bp)
Sequence length: 1359 bp
------------
```

4. Algorithm and implementation

Megraft aims to extend fragments of SSU sequences from e.g. metagenomic data into full-length sequences. This is achieved by using a reference database of full-length 16S/18S rRNA genes (SILVA release 108; Pruesse et al. 2007). Megraft also models variation in both observed and unobserved regions of the SSU sequence according to a sequence variability model based on HMM profiles from V-Xtractor (Hartmann et al. 2010) and Metaxa (Bengtsson et al. 2011). Megraft is able to extend sequences of bacterial, eukaryote, and archaeal origin. It is not specifically adapted for mitochondrial and chloroplast 16S sequences, though it would be possible to use Megraft to extend such sequences by using an appropriate custom database.

Megraft uses NCBI-BLAST (Altschul et al. 1997) to find the most similar reference sequence with respect to each input fragment. It then employs a HMMER (version 3) search (Eddy 2011) to find conserved regions in the reference sequence. Subsequently, the program compares the reference sequence to the input fragment to determine the observed variation in conserved and variable regions, respectively. This comparison is performed based on MAFFT (Katoh and Toh 2008) alignments. After these steps have been performed, Megraft treats each input fragment – reference sequence pair according to the mode specified by the user. The three modes of running Megraft are explained below.

Proxy mode

In Megraft's simplest mode – called the proxy mode – the input fragment is only used to find the best possible reference sequence in the database, and that reference sequence is written to the output file as a proxy for the input fragment. This mode is conservative when it comes to estimating the diversity in a community, but it may underestimate the actual diversity if the community under study contains many organisms that are substantially different from the species present in the reference database.

Insert mode

If Megraft is running in the insert mode, the input fragment is first used to find the most similar reference sequence, just as in the proxy mode. However, when the input fragment has been aligned to the database sequence using MAFFT, the input fragment is inserted into (and replaces that part of) the reference sequence at the appropriate position. This creates an extended sequence that is more similar to the input sequence than the reference sequence was. This extended sequence is then written to the output file. This mode accounts for previously unobserved diversity in a better way than the proxy mode, but it does not account for variation between the sequences outside of the input fragment.

Insert-differential-introduce mode

This mode is Megraft's most sophisticated mode, and it produces the most accurate results in the absolute majority of cases. The insert-differential-introduce mode is based on the insert mode and carries out the same steps. However, this mode also takes unobserved variation into account by applying a model of variability in conserved and variable regions of the SSU sequence. This model is then used to introduce pseudorandom variation into the extended sequence outside of the inserted fragment. This mode accounts for previously unobserved diversity across the full sequence length. It should be noted however that the sequences produced by this mode (as well as the insert mode) are not real sequences observed in nature but rather artificial constructs that should only be used to make more accurate species diversity estimates in metagenomic datasets (and similar data).

While Megraft's default settings should be appropriate in most situations, you should still examine whether they are suitable for your purposes and for the dataset at hand. If the dataset is small, this can be done by running the software multiple times on the data, with different settings, and analyse the outcome. On larger datasets, it might be more feasible to only run Megraft on a sub set of the sequences for testing.

Although we have found the most advanced mode of Megraft to perform better than the less advanced modes is six of the eight studies examined, we recommend that the user examine and compare the output from all three modes in terms of consistency and performance.

We do not recommend using the sequences produced by Megraft for anything else than rarefaction and sequencing depth analysis. The sequences, although derived through a Jukes-Cantor model of sequence evolution, are, at least in part, artificial. The less similar a sequence fragment is to a reference SSU sequence, the more mutations will be added to it, meaning that some sequences will be identical or very similar to reference sequences, whereas others will be more divergent. Clustering and rarefaction-type approaches deal with degree of divergence, a purpose for which the Megraft sequences are fine. Phylogenetic inference, on the other hand, deals both with degree of divergence and the distribution of that divergence over the lineages under scrutiny. The Megraft sequences are not suitable for such purposes.

In many occasions, Megraft would be used as part of a pipeline with a software program for extraction of SSU sequences from a metagenome. To provide a runtime estimate for such a situation, we pooled 1,522 SSU fragments of 350 bp. to a dataset of one million, 350 bp. sequences of random nucleotide data and subjected the joint file to SSU extraction through Metaxa on a 4-core Linux cluster. Afterwards, the output sequences were fed into Megraft and run separately for archaeal, bacterial and eukaryote SSU sequences. The run took 462 minutes in total, with Megraft adding only 55 minutes (12%) to the total running time.

Task	Software	# Sequences	Time taken (min.)
SSU Extraction	Metaxa	1,001,522	407
Grafting of archaeal SSUs	Megraft	19	1
Grafting of bacterial SSUs	Megraft	1,038	42
Grafting of eukaryote SSUs	Megraft	438	12
Total time		1,001,522	462

5. License information

6. References

Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402

Bengtsson J, Eriksson KM, Hartmann M, Wang Z, Shenoy BD, Grelet G-A, Abarenkov K, Petri A, Alm Rosenblad M, Nilsson RH (2011) Metaxa: a software tool for automated detection and discrimination among ribosomal small subunit (12S/16S/18S) sequences of archaea, bacteria, eukaryotes, mitochondria, and chloroplasts in metagenomes and environmental sequencing datasets. Antonie Van Leeuwenhoek 100:471–475. DOI: 10.1007/s10482-011-9598-6

Eddy SR (2011) Accelerated Profile HMM Searches. PLoS Comput Biol 7:e1002195. DOI: 10.1371/journal.pcbi.1002195

Hartmann M, Howes CG, Abarenkov K, Mohn WW, Nilsson RH (2010) V-Xtractor: an open-source, high-throughput software tool to identify and extract hypervariable regions of small subunit (16S/18S) ribosomal RNA gene sequences. J Microbiol Methods 83:250–253. DOI: 10.1016/j.mimet.2010.08.008

Katoh K, Toh H (2008) Recent developments in the MAFFT multiple sequence alignment program. Brief Bioinformatics 9:286–298. DOI: 10.1093/bib/bbn013

Pruesse E, Quast C, Knittel K, Fuchs BM, Ludwig W, Peplies J, Glöckner FO (2007) SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. Nucleic Acids Res 35:7188–7196. DOI: 10.1093/nar/gkm864