September 24, 2014

MetAmp: a tool for Meta-Amplicon analysis
User Manual

Ilya Y. Zhbannikov[1], Janet E. Williams[1], James A. Foster[1,2,3]

[3]Institute for Bioinformatics and Evolutionary Studies, University of Idaho, Moscow, ID 83844-3051,
[2]Department of Biological Sciences, University of Idaho, Moscow, ID 83844-3051,
[1]Department of Bioinformatics and Computational Biology, University of Idaho, Moscow, ID 83844-3051,

Keywords:
Bioinfomatics, metagenomics, 16S genes, microbial studies, data mining

Correspondence:
Ilya Y. Zhbannikov
Department of Bioinformatics and Computational Biology
University of Idaho
Moscow, ID 83844-30521
Email: zhba3458@vandals.uidaho.edu

1 Description

MetAmp tool was developed to analyze microbial amplicon data by combining several marker regions from 16S rRNA gene. Such marker regions serve as unique identifications for species. There are nine marker regions in total in bacterial 16S rRNA gene.

MetAmp was developed by Ilya Y. Zhbannikov (zhba3458@vandals.uidaho.edu, i.zhbannikov@mail.ru), James A. Foster (foster@uidaho.edu) and Janet Williams (janetw@uidaho.edu).

2 Installation

MetAmp is easy to install. However, you must have R (`www.r-project.org`), Python (`www.python.org`) and GCC (`https://gcc.gnu.org/`).

1. Download or clone MetAmp from our GitHub repository: `http://github.com/izhbannikov/MetAmp`

2. In Terminal: `cd` to the MetAmp directory and execute makefile: `make`. First, the installation script will check for required tools: R, Python and GCC. Then it builds and installs required packages and data for evaluation. You can also run `make install` to avoid checking stage.

3 Quick start

1. Edit script main.R, specifically:
   - Provide your sequence library (amplicon sequence data), for example:

```
libs <- c("<MetAmp directory>/Evaluation/data/staggered/SRR072221_forward.fastq", # V1-3
          "<MetAmp directory>/Evaluation/data/staggered/SRR072237_forward.fastq", # V3-5
          "<MetAmp directory>/Evaluation/data/staggered/SRR072236_forward.fastq") # V6-9
```

   - Provide reference sequences, for example:

```
# Reference 16S gene sequences
ref16S <- "Evaluation/data/16S.fasta"
# V1-3 # Reference gude (marker) regions
refs <- c("Evaluation/data/V13.fasta",
          "Evaluation/data/V35.fasta", # V3-5
          "Evaluation/data/V69.fasta") # V6-9
```

2. Set the program directory in `main.R`:

```
dir_path <- "~/Projects/metamp/"
```

3. Run the script `main.R`:

```
source("main.R")
```

3.1 How to run the program on test data

The following test data sets were used:

- Evaluation/data/even
- Evaluation/data/staggered

The even and staggered are the Human mock community pyrosequence data (SRX021555), even and staggered community, 20 species. To be able to use these sets you need to unzip them into any folder you want.

4 MetAmp workflow

In the following sections we present key stages of our meta-amplicon analysis pipeline.

4.1 Input data

Input data can be any of the following NGS libraries: Roche 454, Ion Torrent, Illumina.

Roche 454 and Ion Torrent provide single-end sequences in special binary Sequence Flowgram Format (SFF). Those reads are ready for using them in downstream analysis.

Illumina paired-end reads should be merged into longer single-end sequences. For this task you can use FLASH program that provided by Magos and Salzberg (2011).

4.2 Data denoising

Data denoising is necessary for downstream analysis because it removes the majority of 'foreign' nucleotides, such as barcodes and primers, and low-quality regions. We suggest you to use SeqyClean tool for this operation, since it is the most comprehensive tool for sequence denoising. You can download SeqyClean here: `https://bitbucket.org/izhbannikov/seqyclean`. Using SeqyClean is simple and straightforward. Nevertheless, we provided a special script that does all the data de-noising.

4.3 Analysis workflow

1. Building a reference topology of microbial populations,where pairwise distances are computed from applying pairwise alignment of complete reference 16S gene sequences from RDP database, see Figure 1(a), top plane. We use the percent sequence identity in order to compute the distances between sequences.

2. Computing a guide "amplicon" topology for each reference 16S sequence. To do this, we extract marker regions (for example V1- V3 and V6-V9) from reference complete 16S sequences ("reference points") and place them on to the plane with the same methodology that was used for building the reference topology of complete 16S (see Figure 1(a), bottom plane,

hollow green circles).

3. add empirical amplicon sequences (in fact, amplicon consensus sequences), obtained through amplicon sequencing of microbial data (Figure 1b, bottom, filled blue circles). This topology is the same empirical topology as if the empirical data would be clustered with existing methods

4. Normalized empirical topologies are formed foreach marker through mapping of each empirical point back to the reference 16S sequence topology. Mapping between the reference topology and guide "amplicon" topology is achieved with affine transformation. Guide sequences are so match with the corresponding reference 16S sequences, from where they were obtained from (see Figure 1a, top, solid green circles), and, in turn, carry the empirical amplicon sequences (or consensus amplicon sequences) back to the 16S plane.

5. Repeating stages 2-4 for each variable region.

6. Clustering and building OTUs.

7. Final statistics.

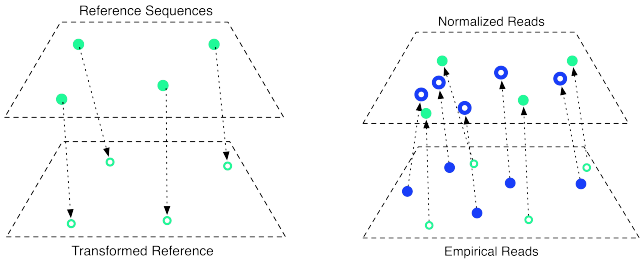
4.4 Output files

The results contain:

- Clustering results (`*.clust`) - by default it is clusters.clstr but the name can be changed (by editing `final_clust_filename` parameter in `config.R` file)

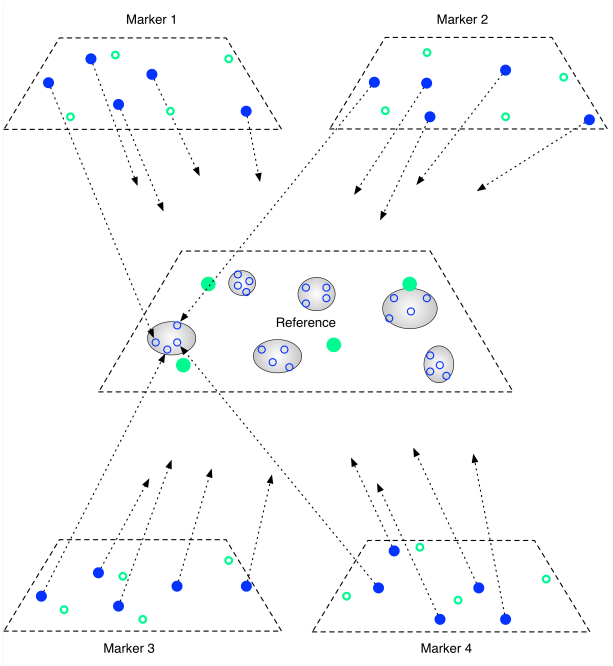
- A text file that contain coordinates for each point, including reference (`coordinates.crd`)

- A text file containing all messages during the analysis process (`log.txt`)



(a)

(b)



(c)

Figure 1: Ilustration of "meta-amplicon" analysis algorithm. (a) Computing reference (top plane) and guide (bottom plane) topologies. Pairwise sequence dissimilarity is used to compute distances between reference sequences. The same approach is used for guide sequences, originated from corresponding whole 16S gene sequences. Reference and guide points are then placed on to the planes with multidimensional scaling. (b) The empirical amplicon sequences (hollow blue circles) are places on to the bottom plane along with the guide circles (hollow green) circles and with the same metric. Such guide sequences are then mapped back to the reference 16S plane, carrying empirical sequences (solid blue circles). (c) Then this is repeated for each of marker (variable) region.

5 Provided data

We provided some data that can be used in your projects. Notice that the amount of data is very large (even unzipped files can take several Gb). For evaluation purposes we provided small data set. Description of data uploaded to our storage is given below:

- LTP folder contains data (16S sequences, extracted marker regions and distance matrices) for data extracted from All Species Living Tree Project.

  LTP:

Name	Description
LTP-10271.fasta	Whole reference 16S sequences
LTP-10271_V13.fasta	reference sequences for marker 1-3
LTP-10271_V35.fasta	sequences for marker 3-5
LTP-10271_V69.fasta	sequences for marker 6-9

- RDP folder contains the same type of data but extracted from Ribosomal Database Project: http://rdp.cme.msu.edu/

  RDP:

Name	Description
refSetV13-100.fasta	reference sequences for marker 1-3
refSetV35-100.fasta	reference sequences for marker 3-5
refSetV69-100.fasta	reference sequences for marker 6-9
refSet16S-100.fasta	whole 16S reference sequences

- Evaluation data set contains a smal set of reference sequences (100 sequences), amplicon sequences and pre-clustered data sets. Amplicon sequences contain sequence data from Human Mock Community. More information provided here: `http://www.hmpdacc.org/HMMC/`

  Evaluation:

Name	Description
even	sequence data for even community (ids)
staggered	sequence data for staggered community (ids)
16S.fasta	contains 30 reference 16S sequences
V13.fasta	contains 30 reference marker sequences (1-3 regions)
V35.fasta	contains 30 reference marker sequences (3-5 regions)
V69.fasta	contains 30 reference marker sequences (6-9 regions)

The even and staggered are the Human mock communuty (HMP) pyrosequence data (SRX021555: `http://www.ncbi.nlm.nih.gov/sra?term=SRX021555`). Detailed description of these datasets (and sequencing protocols) is under the following link: `http://www.hmpdacc.org/HMMC/`

To run the program on test data open `evaluation.R` and set the working directory (this directory should contain test data directory) and run the script `evaluation.R` from R-environment:

```
>source("evaluation.R")
```

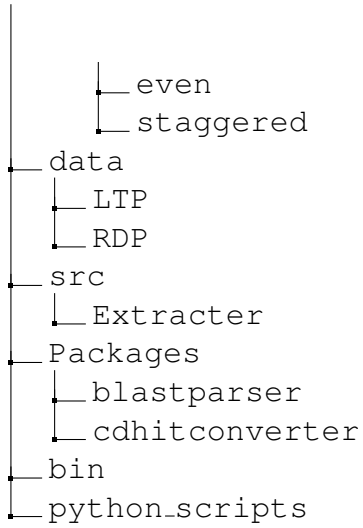
By default we provided pre-computed datasets that contain:

- Distance matrix (and actual reference sequences in FASTA format) computed out of reference 16S genes. These reference sequences were downloaded from All Species Living Tree Project web site(`http://www.arb-silva.de`). There are 10,271 sequences in total.
- Distance matrices from three (V1-3, V3-5 and V6-9) marker regions extracted from the reference set described above. So in total, there are 10,271 sequences for each marker region.

Inside of the program directory we also provided some data.

5.0.1 Directory map and descriptions of folders

```
MetAmp
└── Evaluation
    └── data
```



Evaluation folder contains data for evaluation purposes.

Evaluation/data - contains sequence libraries for evaluation: even and staggered communities and evaluation scripts.

data - contains reference sequences and other data that can potentially be useful

data/LTP - contains reference sequences and pre-computed distance matrices for $> 10,000$ species from All Species Living Tree Project.

data/RDP - contains reference sequences and pre-computed distance matrices for 100 species from Ribosomal Database Project.

src - contains source files.

Packages - contains installation files.

Packages/blastparser - a parser for blastn files.

Packages/cdhitconverter - converts CD-HIT's .clstr data into tabular format.

bin - binary files (alignment and clustering programs). **python_scripts** - Python scripts downloaded from `www.drive5.com`.

5.1 Hints on running your meta-amplicon analysis

Before analysis, you may need to perform data de-noising and (if you use Illumina sequence data), merge overlapping reads. These have been already described in the section above.

Later I will provide the scripts that do it.

1. Edit script `config.R`, specifically:
   Provide your sequences (**Note:** your sequences should be in forward 5'-3' direction!), for example:

```
libs <- c("Evaluation/data/staggered/SRR072221_forward.fastq", # V1-3
          "Evaluation/data/staggered/SRR072237_forward.fastq", # V3-5
          "Evaluation/data/staggered/SRR072236_forward.fastq") # V6-9
```

   and reference sequences:

```
# Reference 16S gene sequences
ref16S <- "Evaluation/data/16S.fasta"
# Reference gude (marker) regions
refs <- c("Evaluation/data/V13.fasta", # V1-3
          "Evaluation/data/V35.fasta", # V3-5
          "Evaluation/data/V69.fasta") # V6-9
```

2. Set the program directory in `main.R`, for example:

```
dir_path <- "~/Projects/metamp/"
```

3. Run the script `main.R`:

```
source("main.R")
```

6 Parameters

Here we describe **config.R**:

analysis_dir - contains analysis files.

ref16S - path to the file with reference 16S sequences (file must be in FASTA format)

refs - path to the file with guide sequences, i.e. variable regions (file must be in FASTA format)

libs - data (empirical libraries)

denoise_app - path to sequence denoising application (by default - USEARCH)

merge_app - path to application that merges overlapping paired-end reads (by default - FLASH)

cluster_app - clustering application (by default - USEARCH). Clustering is used to reduce amount of data by computing consensus sequences.

cluster_identity_threshold - identity threshold for clustering application (by default - 0.9)

cluster_suff - suffix for clustered intermediate analysis files

7 Acknowledgements

This work was made possible by NIH Grants P20GM16448 and P20GM16448, and NSF Grant DBI0939454.