AcetoScan

(version 1.0)

Singh, Abhijeet, Johan A. A. Nylander, Anna Schnürer, and Bettina Müller. 2020. High Throughput Sequencing and Automated Analysis of Formyltetrahydrofolate Synthetase (FTHFS) Gene Amplicons to Estimate Acetogenic Population Dynamics.

Contents

AcetoScan

- Version: 1.0 (20200413)
- Last modified: Apr 13, 2020 20:30
- Sign: Abhijeet Singh (abhijeetsingh.aau@gmail.com)

Overview

AcetoScan is a software pipeline for the unsupervised analysis of high-throughput sequencing data of the formyltetrahydrofolate synthetase (FTHFS) gene amplicon sequencing. The pipeline is primarily designed for the analysis of data generated on Illumina MiSeq platform, however, it could potentially be used for the sequence data generated on other sequencing platforms. AcetoScan can also process fasta sequence data to filter out non-target sequences, assign taxonomy and generate phylogenetic tree.

Dependencies

AcetoScan is built on following dependencies and requires the software versions equals to or higher than the mentioned versions:

- Cutadapt v1.18-1 (Martin 2017)
- VSEARCH v2.13.1 (Rognes et al. 2016)
- NCBI-blast+ v2.5.0+ (Camacho et al. 2009)
- Bioperl v1.7.2-3 (Jamison 2005)
- MAFFT v7.307 (Katoh and Standley 2013)
- Fasttree 2.1.9 (Price, Dehal, and Arkin 2009)
- AcetoBase (Singh et al. 2019)
- R v3.5.2 (R Core Team 2011)
    - Phyloseq v1.24.2 (McMurdie and Holmes 2013)
    - ggplot2 v3.1.1 (Ginestet 2011)
    - plotly v4.9.0 (Sievert 2018)
    - RcolorBrewer v1.1.2 (Neuwirth 2014)
    - plyr v1.8.4 (Wickham 2014)
    - dplyr v0.8.0.1 (Wickham and Francois 2016)
    - vegan (Oksanen et al. 2019)

Installation

AcetoScan version v1.0 is install compatible for Debian/Ubuntu based systems. Other system specific methods can be used to install the dependency software followed by AcetoScan installation. To install AcetoScan and its dependencies on Debian/Ubuntu based systems following methods can be used:

Installing AcetoScan as a super user:

```
$ sudo ./INSTALL

$ acetoscan -h
```

Installing as a super user place the AcetoScan scripts in path `/usr/local/bin/` and will create directory `acetoscan` in path `/home/<user>/`, and three sub-directories `acetoscan_bin` containing dependency binaries, `acetobase` containing AcetoBase reference protein database and `output_data` as a default output directory for `acetoscan` analysis.

Installing AcetoScan as a local user:

```
$ bash INSTALL

$ bash /home/<user>/acetoscan/acetoscan -h
```

If the installation needs to be done as normal user (without super user privilege), installation will create directory `acetoscan` in path `/home/<user>/`, and three sub-directories `acetoscan_bin` containing dependency scripts, `acetobase` containing AcetoBase reference protein database and `output_data` as a default output directory for `acetoscan` analysis. It is to be noted that some software dependencies needs to be installed manually if they are not already installed or if installation of AcetoScan is not done as super user.

Prerequisites

To run AcetoScan analysis, the of the raw data files must be present in input directory or sub-directories in the compressed fastq format (Illumina Inc. 2015). The file name can contain several fields *i.e.* `Sample_<name>_<batch>_<date>_L001_R1_001.fastq.gz|bz2`, however, it is recommended to use short names *i.e.* `Sx_<date>_L001_R1_001.fastq.gz` wherever possible for better visualization in the output plots.

The quality of the raw data should be visualized by the FastQC and MiltiQC analysis. Samples which do not appear to be well sequenced in terms of number of reads, extremely bad sequence quality *et cetera* should be removed before starting the analysis.

Some analysis only requires FTHFS sequences in multifasta format with no special requirements or modification. These analysis are further discussed in following text.

Description of analysis

Based on the information available for the published primer pairs targeting partial FTHFS gene sequence, most of the primers generate amplicons greater than 600 base pairs (Leaphart and Lovell 2001; Ohashi et al. 2007; Müller, Sun, and Schnürer 2013; Müller et al. 2016). So, the sequence generated in Illumina MiSeq sequencing cannot be merged. Therefore, AcetoScan processes the sequence data only for one type of reads as specified by the user (default = R1/forward reads). AcetoScan carries out unsupervised data analysis in four major steps and results into high quality and interactive plots. The analysis steps are further described below:

Step 1. Quality control

In this step, the user specified raw sequence data (either forward or reverse read data) is subjected to adapter/primer sequence trimming and quality filtering. This step is dependent on software cutadapt version ≥1.18. The trimming is done by cutting specified number of bases (default = 24) from the 5' end of the fastq sequences. The number of bases trimmed can be changed based on the length of the primer sequence. Filtering of the sequences is done according to the specified Phred quality score threshold (default =20). In general, the sequence data having Phred quality score of ≥ 20 (99 % base call accuracy) is considered good enough for the sequence analysis, therefore set as default. The sequences are also filtered based on the minimum (default = 120) and maximum lengths (default = 300) of the fastq sequnces. The options can be changed by the `acetoscan` options presented in table 1.

Step 2. Sequence analysis

The second step is sub-divided into three sub-analysis process which require the dependency software VSEARCH version ≥2.13.1. The description of the analysis is described below:

I. Data curation

a. Dereplication

Dereplication of the sequence data refers to the removal of redundancies in the sequences and only keeping absolute unique sequences based on complete length of sequence and discarding the duplicate sequences. For dereplication step, multithreading is not supported. Dereplication is done by the VSEARCH command `--derep_fulllength`. The minimum clustering

threshold for dereplication is set to 2, to remove the sequences which appear only once in whole data being analysed.

b. Denoising

The sequence clustering and denoising is done based on the UNOISE algorithm version 3 implemented in VSEARCH command `--cluster_unoise`. In this step the reads with sequencing and PCR error are removed and biologically correct sequences are retained and clusters of sequences are generated. The minimum cluster size is set to 2 as default. User can define the minimum cluster size using `acetoscan` option -c.

c. Chimera filtering

Chimera sequences are removed from the denoised data using default values from VSEARCH command `--uchime3_denovo` based on the UCHIME2 algorithm. The non-chimera sequences are further used for the OTU picking.

d. OTU picking

Operational taxonomic units (OTU) are generated based on the minimum cluster size and minimum cluster threshold parameters specified by the user. The default parameters for the clustering threshold is set to 80 % for the genus level resolution (acetobase 2019).

II. Data filtering

At this point, the non-target sequences *i.e.* sequences which are not FTHFS sequences are filtered out. The removal of sequences is done with blastx algorithm using AcetoBase reference protein database. The sequence filter is done based on the evalue criteria which is set to 1e-3 as default, however, for increased accuracy user can change the evalue. Further, after discarding the non-targeted sequences, the retained FTHFS sequences are analysed for the longest reading frame without internal stop codons. This is done by AcetoScan using Bioperl as base package (bioperl).

III. Generation of Abundance and taxonomic tables

The longest-best FTHFS sequences are used for the OTU table generation with the clustering threshold chosen in the OTU picking step. VSEARCH command `--usearch_global` generates the abundance table. Taxonomy table is generated by the frame checked FTHFS sequences against the AcetoBase protein database using blastx algorithm with e-value criteria mentioned above.

Step 3. Sequence alignment and phylogenetic inference

The FTHFS OTU sequences in correct frame of codons are saved in fasta formatted file. These sequences are used for the global sequence alignment with MAFFT (mafft citation) aligner with 5 rounds of UPGMA tree refinement and iteration specified by the user. A good alignment facilitates faster phylogenetic tree generations, therefore the iteration cycles for the multiple sequence alignment and phylogenetic bootstrap iterations are set as common parameter for both the steps and can be specified by the -B option (table 1). The default number of iterations in multiple sequence alignment and bootstraps in tree building is set to 1000 as default. Phylogenetic tree with the aligned sequences is generated with software Fasttree with Jukes-Cantor (JC) distance and maximum likelihood (ML) topology refinement with 10 rounds of nearest neighbor interchange (NNI) and 5 rounds of subtree pruning and re-grafting (SPR). The generalized time reversible (GTR) model with 10 rounds of CAT approximation and GAMMA rate heterogeneity and BIONJ distance optimization together with specified bootstrap iterations is applied for the phylogenetic tree generation.

Step 4. Data visualization

The final data generated in step 2 and three are used for the data visualization in form of various plots. The visualization of data is done with R environment where abundance table and taxonomy table are merged together with the phylogenetic tree and sample table to generate a phyloseq object with the package *phyloseq*. For the data table modification and diversity analysis package *plyr*, *dplyr* and *vegan* are used. Barplots and heatmaps for all taxonomic levels (phylum to species) are generated with different abundance threshold in visualization which is specified in the respective plot. Alpha diversity analysis is done with package *phyloseq* for Observed, Shannon and Simpson diversity measure indices. Beta diversity is visualized in form of non-metric multidimensional scaling (NMDS) and principal coordinates analysis (PCoA) analysis. For NMDS analysis is carried out with the Bray-Curtis dissimilarity distances and plotted in two different forms *i.e.* based on the phyla and sample. PCoA analysis (also known as multidimensional scaling - MDS) is carried out with the weighted UniFrac distances. The plots in elegant and publication ready format were generated with package *ggplot2* and *RColorBrewer*. Interactive plots in html format to visualize in web-browser were are with the package *plotly*. Phylogenetic tree visualization and annotation is done at the phylum level.

AcetoScan pipeline

AcetoScan pipeline is primarily developed for the unsupervised analysis and visualization of the raw sequencing data in compressed fastq format. However, if user want to process FTHFS sequence data which is in fasta format and is generated by clone library construction and Sanger sequencing, AcetoScan harbors functionalities for this. AcetoScan pipeline has four different analysis programs i.e. `acetoscan`, `acetocheck`, `acetotax` and `acetotree`. These program executable scripts will be installed in directory based on the method of installation as discussed above. The programs available in AcetoScan pipeline can be seen by following command

```
$ acetoscan -X

# AcetoScan commands:

  acetoscan  - for complete processing of raw illumina MiSeq output data
  acetocheck - for processing fasta sequences and filtering out non-target sequences
  acetotax   - acetocheck + taxonomic assignments
  acetotree  - acetotax + phylogenetic tree generation
```

1. acetoscan

It is the main program of AcetoScan pipeline and it requires the raw sequence data in compressed fastq format and results into ready to use graphs and plots. This program require the user to give the path to the directory containing the raw sequence data. The only prerequisite of `acetoscan` is the compressed fastq format as discussed above. The options for `acetoscan` can be seen using help.

```
$ acetoscan -h

Example: bash /home/abhi/acetoscan/acetoscan -i /<input path>/ -o /<output path>/
           -m 300 -n 120 -q 20 -l 24 -r 1 -t 0.80 -c 2 -e 1e-3 -B 1000 -P 8


        -i      Input directory containing raw illumina data
        -o      Output directory
                        :default = /home/abhi/acetoscan/output_data
        -m      Maximum length of sequence after quality filtering
                        :defalut max_length = 300
        -n      Minimum length of sequence after quality filtering
                        :defalut min_length = 120
        -q      Quality threshold for the sequences
                        :default quality threshold = 20
        -l      Primer length
                        :default primer length = 24
        -r      Read type either forward or reverse reads
                        1 = forward reads (default), 2 = reverse reads
        -t      Clustering threshold
                        :default cluster threshold = 0.80 (80 %)
        -c      Minimum cluster size
                        :default minimum cluster size = 2
        -e      E-value
                        :default evalue = 1e-3
        -B      Bootstrap value
                        :default bootstrap = 1000
        -P      Parallel processes / threads
                        :default no. of parallels = all available threads
        -h      Print help
        -X      Print AcetoScan commands
        -v      Print AcetoScan version
        -C      Print AcetoScan citation
```

Running acetoscan on AcetoScan test data:

```
$ acetoscan -i /home/<user>/Desktop/test_data -o /home/<user>/Desktop/test_result
```

Program `acetoscan` will result into two directories **a)** `output_data` – in this directory the data processing files will be located. In case of the execution halt or process failure, the data can be accessed for this or sub-directories, **b)** `acetoscan_result` – which contains the 67 final result files after successful execution of `acetoscan`. The file generated as results by `acetoscan` is presented in following tree:

```
/<path_to_output_directory>/acetoscan_result/
│
├── 0_acetoscan_<date>_<time>.log
├── 0_visualization_info.txt
├── 1_Phylum_abs_abundance.html
├── 1_Phylum_abs_abundance.pdf
├── 1_Phylum_abs_abundance.tif
├── 1_Phylum_barplot.html
├── 1_Phylum_barplot.pdf
├── 1_Phylum_barplot.tif
├── 2_Class_barplot.html
├── 2_Class_barplot.pdf
├── 2_Class_barplot.tif
├── 3_Order_barplot.html
├── 3_Order_barplot.pdf
├── 3_Order_barplot.tif
├── 4_Family_barplot.html
├── 4_Family_barplot.pdf
├── 4_Family_barplot.tif
├── 4_Family_heatmap.html
├── 4_Family_heatmap.pdf
├── 4_Family_heatmap.tif
├── 5_Genus_barplot.html
├── 5_Genus_barplot.pdf
├── 5_Genus_barplot.tif
├── 5_Genus_heatmap.html
├── 5_Genus_heatmap.pdf
├── 5_Genus_heatmap.tif
├── 6_Species_barplot.html
├── 6_Species_barplot.pdf
├── 6_Species_barplot.tif
├── 6_Species_heatmap.html
├── 6_Species_heatmap.pdf
├── 6_Species_heatmap.tif
├── 7_Absolute_abundance.pdf
├── 8_Relative_abundance.pdf
├── Alpha_diversity.html
├── Alpha_diversity.pdf
├── Alpha_diversity.tif
├── FTHFS_otu.aln
├── FTHFS_otu.fasta
├── FTHFS_otutab.csv
```

```
├── FTHFS_otu.tree
├── FTHFS_samtab.csv
├── FTHFS_taxtab.csv
├── FTHFS_tree1.html
├── FTHFS_tree1.pdf
├── FTHFS_tree1.tif
├── FTHFS_tree2.html
├── FTHFS_tree2.pdf
├── FTHFS_tree2.tif
├── NMDS_Phylum_1.html
├── NMDS_Phylum_1.pdf
├── NMDS_Phylum_1.tif
├── NMDS_Phylum_2.html
├── NMDS_Phylum_2.pdf
├── NMDS_Phylum_2.tif
├── NMDS_Sample_1.html
├── NMDS_Sample_1.pdf
├── NMDS_Sample_1.tif
├── NMDS_Sample_2.html
├── NMDS_Sample_2.pdf
├── NMDS_Sample_2.tif
├── weighted_unifrac_PCoA_2.html
├── weighted_unifrac_PCoA_2.pdf
├── weighted_unifrac_PCoA_2.tif
├── weighted_unifrac_PCoA.html
├── weighted_unifrac_PCoA.pdf
└── weighted_unifrac_PCoA.tif
```

2. acetocheck

`acetocheck` can be used to filter out the FTHFS sequences, discard non-FTHFS sequences and check FTHFS reading for internal stop codons. Only the true FTHFS sequence frame without internal stop codon is selected and written out to output file. `acetocheck` can be with user specific evalue for filtering the sequences, however, the default evalue is set to 1e-3.

```
$ acetocheck -h

Example: bash /home/abhi/acetoscan/acetocheck -i /path/<input file> -o /path/<output
          file> -e 1e-3 -P 8


        -i      Input_file - multifasta file with squences
        -o      Output file
                        :default =  acetocheck_<DATE>_<TIME>.fasta
        -e      E-value
                        :default evalue = 1e-3
        -P      Parallel processes/threads
                        :default no. of parallels = all available threads
        -h      Print help
        -X      Print AcetoScan commands
        -v      Print AcetoScan version
        -C      Print AcetoScan citation
```

3. acetotax

`acetotax` has two sub-processing step which includes `acetocheck` as first step followed by taxonomic assignments of the filtered sequences. Filtering and taxonomic assignment threshold can be changed according to the user specific parameters.

```
$ acetotax -h

Example: bash /home/abhi/acetoscan/acetotax -i /path/<input file> -o /path/<output
          file> -e 1e-3 -P 8


        -i      Input_file - multifasta file with squences
        -o      Output file
                        :default =  acetotax_<DATE>_<TIME>.fasta
                        :default = acetotax_<DATE>_<TIME>.csv
        -e      E-value
                        :default evalue = 1e-3
        -P      Parallel processes/threads
                        :default no. of parallels = all available threads
        -h      Print help
        -X      Print AcetoScan commands
        -v      Print AcetoScan version
        -C      Print AcetoScan citation
```

4. acetotree

`acetotree` is the program to generate the phylogenetic tree from the FTHFS sequences. `acetotree` is based on `acetotax` followed by the multiple sequence alignment of the FTHFS squences and phylogenetic tree generation. The bootstrap is the common iteration value for the multiple sequence alignment and the phylogenetic tree construction.

```
$ acetotree -h


Example: /home/abhi/acetoscan/acetotree -i /path/<input file> -o /path/<output file>
          -e 1e-3 –B 1000 -P 8


        -i      Input_file - multifasta file with squences
        -o      Output file
                        :default =  acetotree_<DATE>_<TIME>.fasta
                        :default = acetotree_<DATE>_<TIME>.csv
                        :default = acetotree_<DATE>_<TIME>.aln
                        :default = acetotree_<DATE>_<TIME>.tree
        -e      E-value
                        :default evalue = 1e-3
        -B      Bootstrap value
                        :default bootstrap = 1000
        -P      Parallel processes/threads
                        :default no. of parallels = all available threads
        -h      Print help
        -X      Print AcetoScan commands
        -v      Print AcetoScan version
        -C      Print AcetoScan citation
```

Bibliography

Camacho, Christiam, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L. Madden. 2009. "BLAST+: Architecture and Applications." *BMC Bioinformatics*. https://doi.org/10.1186/1471-2105-10-421.

Ginestet, Cedric. 2011. "Ggplot2: Elegant Graphics for Data Analysis." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. https://doi.org/10.1111/j.1467-985x.2010.00676_9.x.

Illumina Inc. 2015. "MiSeq Reporter Generate FASTQ: Workflow Guide." San Diego, California 92122 U.S.A.: Illumina, Inc. https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/miseqreporter/miseq-reporter-generate-fastq-workflow-guide-15042322-01.pdf.

Jamison, D. Curtis. 2005. "Bioperl." In *Perl Programming for Biologists*. https://doi.org/10.1002/047172274x.ch11.

Katoh, Kazutaka, and Daron M. Standley. 2013. "MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability." *Molecular Biology and Evolution*. https://doi.org/10.1093/molbev/mst010.

Leaphart, Adam B, and Charles R Lovell. 2001. "Recovery and Analysis of Formyltetrahydrofolate Synthetase Gene Sequences from Natural Populations of Acetogenic Bacteria." *Applied and Environmental Microbiology* 67 (3): 1392–95. https://doi.org/10.1128/AEM.67.3.1392.

Martin, Marcel. 2017. "Cutadapt 1.13."

McMurdie, Paul J., and Susan Holmes. 2013. "Phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data." *PLoS ONE*. https://doi.org/10.1371/journal.pone.0061217.

Müller, Bettina, Li Sun, and Anna Schnürer. 2013. "First Insights into the Syntrophic Acetate-Oxidizing Bacteria - a Genetic Study." *MicrobiologyOpen*. https://doi.org/10.1002/mbo3.50.

Müller, Bettina, Li Sun, Maria Westerholm, and Anna Schnürer. 2016. "Bacterial Community Composition and Fhs Profiles of Low- and High-Ammonia Biogas Digesters Reveal Novel Syntrophic Acetate-Oxidising Bacteria." *Biotechnology for Biofuels* 9 (1): 1–18. https://doi.org/10.1186/s13068-016-0454-9.

Neuwirth, Erich. 2014. "RColorBrewer: ColorBrewer Palettes." *R Package Version 1.1-2*.

Ohashi, Yuji, Tomoko Igarashi, Fumi Kumazawa, and Tomohiko Fujisawa. 2007. "Analysis of Acetogenic Bacteria in Human Feces with Formyltetrahydrofolate Synthetase Sequences." *Bioscience and Microflora*. https://doi.org/10.12938/bifidus.26.37.

Oksanen, Jari, F Guillaume Blanchet, Michael Friendly, Roeland Kindt, Pierre Legendre, Dan McGlinn, Peter R Minchin, et al. 2019. "Vegan: Community Ecology Package." https://cran.r-project.org/package=vegan.

Price, Morgan N., Paramvir S. Dehal, and Adam P. Arkin. 2009. "Fasttree: Computing Large Minimum Evolution Trees with Profiles Instead of a Distance Matrix." *Molecular Biology and Evolution*. https://doi.org/10.1093/molbev/msp077.

R Core Team. 2011. *R: A Language and Environment for Statistical Computing*. *R Foundation for Statistical Computing*. https://doi.org/10.1007/978-3-540-74686-7.

Rognes, Torbjørn, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. 2016. "VSEARCH: A Versatile Open Source Tool for Metagenomics." *PeerJ* 4 (October): e2584. https://doi.org/10.7717/peerj.2584.

Sievert, Carson. 2018. "Plotly for R." https://plotly-r.com.

Singh, Abhijeet, Bettina Müller, Hans Henrik Fuxelius, and Anna Schnürer. 2019. "AcetoBase: A Functional Gene Repository and Database for Formyltetrahydrofolate Synthetase Sequences." *Database : The Journal of Biological Databases and Curation*. https://doi.org/10.1093/database/baz142.

Wickham, Hadley. 2014. "R: Plyr." *CRAN*.

Wickham, Hadley, and Romain Francois. 2016. "The Dplyr Package." *R Core Team*.