This document describes the installation and use of FunFrame. The colored fonts in this pdf file are clickable hyperlinks.

1 Legal

2 Overview of FunFrame

FunFrame is a pipeline for the ecological analysis of protein-coding genes using targeted metagemomic DNA sequencing. Briefly, FunFrame reads targeted metagenomic sequence data from multiple biological samples, identifies and repairs likely homopolymer read errors, performs ecological analysis on the de-noised sequences, and produces reports describing the microbial communities at each biological sample. The input metagenomic sequence data is typically the product of PCR amplification using universal primers for a specific protein-coding gene prefixed with adaptor and barcode extensions, and read with next-generation pyrosequencing technology.

FunFrame builds upon R [R Core Team, 2012], HMM-FRAME [Zhang and Sun, 2011], UCHIME [Edgar et al., 2011], ESPRIT-Tree [Cai and Sun, 2011], Vegan [Oksanen et al., 2011], and BioPython [Cock et al., 2009]. Optionally, to compute UniFrac distances [Hamady et al., 2010] and subsampled alpha diversity estimates [Gihring et al., 2012], FunFrame uses QIIME [Caporaso et al., 2010b], PyNAST [Caporaso et al., 2010a] and FastTree [Price et al., 2010].

A single bash script runs the full pipeline; alternatively, you can manually run the individual stages of the pipeline. A user-customizable configuration file specifies all parameter settings. Figure 1 sketches the data and programs in FunFrame.

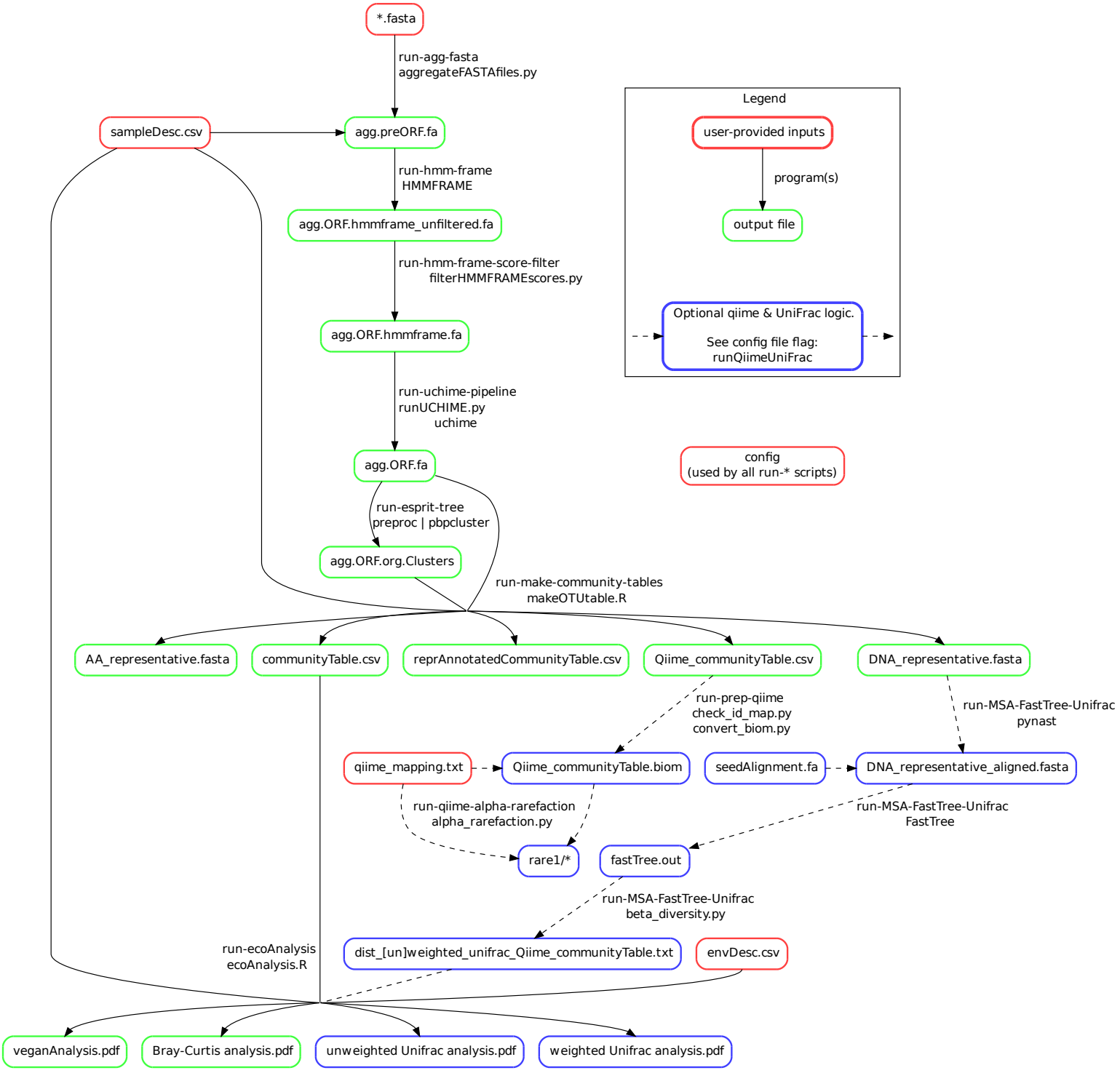

Figure 1: Overview of data flow in FunFrame. The file `flow.pdf` contains a full-sized image. Red vertices are user-provided input files; green represents intermediate files and outputs; edge labels indicate programs. Blue vertices and dashed lines indicate optional QIIME processing for UniFrac distances and sub-sampled alpha diversity estimation.

Listing 1: Installation of additional R libraries.

```
## Install R package dependencies for FunFrame

## install from CRAN
packageNames <- c('ggplot2',
                  'gplots',
                  'optparse',
                  'plyr',
                  'vegan')

install.packages(packageNames)


## install from Bioconductor
source('http://bioconductor.org/biocLite.R')
biocLite('Biostrings')
```

Table 1: R and package dependencies.

Package	**Tested version**
R	2.15.1
Biostrings	2.24.1
ggplot2	0.9.2.1
gplots	2.11.0
optparse	0.9.5
plyr	1.7.1
vegan	2.0-5

3 Installation and Dependencies

FunFrame requires R and several additional R packages. Table 1 lists these requirements and the versions tested with FunFrame.

R can be installed from `http://www.r-project.org/`, and the necessary R packages can be installed by evaluating the file `install_R_package_dependencies.R`, shown in Listing 1.

FunFrame requires Python 2.6.6 or later, which is available at `http://python.org/`, as well as BioPython 1.58 or later, available at `http://biopython.org/wiki/Biopython`.

FunFrame requires HMM-FRAME [Zhang and Sun, 2011] and was tested with version HMMFRAME_3_16; requires UCHIME [Edgar et al., 2011] and was tested with version uchime4.2.40; and requires ESPRIT-Tree [Cai and Sun, 2011] and was tested with version

ESPRITTree11152011.

If you want to compute UniFrac metrics and subsampled alpha diversity, you need to install QIIME which is available at http://qiime.org/. FunFrame has been tested with `QIIME-1.5.0-amd64.vdi` running in VirtualBox on top of Debian Linux 6.

4 Inputs and configuration

FunFrame reads several user-provided input files detailed in this Section (Figure 1).

The FunFrame distribution contains a fully-operational set of input files in the `testData` subdirectory, as well as the expected FunFrame results. These files are useful for validating that the installation is complete and the dependencies are satisfied, and, for understanding how to re-configure FunFrame for new scientific experiments.

4.1 Input FASTA files

The input sequence data is contained in one or more FASTA files, with each file corresponding an environmental sample.

These FASTA files are the products of upstream processing typically including: Converting the raw sequencing output (e.g., sff files) into FASTA; quality filtering; trimming immediately after the barcode sequence; and trimming the 3' ends for consistent length throughout the data set. This processing depends largely on the particular sequencing technology and barcoding conventions, and are addressed in upstream tools such as Mothur [Schloss et al., 2009] and QIIME [Caporaso et al., 2010b].

The FASTA files read by FunFrame are assumed to be trimmed after the barcode sequences, and free of poor-quality `N` nucleotide reads.

The FASTA headers ("> ...") in these files must be present but the header contents are ignored.

4.2 Sample description file

The sample description file `sampleDesc.csv` associates the names of the input FASTA files with their short names used in graphical outputs, as well as with a grouping name.

Following is an example of a sample description file:

```
inputFastaFileName,shortSampleName,group
MarshPlot3_control_plot.fa,m3co,env
MarshPlot7_control_plot.fa,m7co,env
SHC2CnS1F6R01_control_seq.fa,clone,clone
```

Because the file format is comma-separated values (csv), you can use a spreadsheet or text editor to view and change the file. The first line containing the column headers must not be changed.

The group column is used in `ecoAnalysis.R` to group multiple rarefaction curves together, and to assign colors in the PCoA ordination plots. See the sample output `testData/veganAnalysis.pdf` and `testData/Bray-Curtis analysis.pdf` for examples of grouping.

4.3 Main configuration file

The main FunFrame configuration file specifies directories, files, and adjustable parameters. See `testData/config` for an example.

We recommend that you use this file as a template rather than writing a *de novo* configuration file. The sample configuration file is heavily commented and straightforward to modify.

The configuration file is interpreted by bash, so you can include any valid bash expressions or commands in the file. FunFrame evaluates this file multiple times during a full run, so any additional commands you choose typically will be idempotent.

One important item in the configuration is the Boolean flag `runQiimeUniFrac` that governs whether QIIME/UniFrac is to be run. The blue nodes and dotted lines in Figure 1 show the consequences of this flag.

4.4 Ecological variables file

For performing constrained ordination in `ecoAnalysis.R`, the ecological variables file associates short sample names (Subsection 4.2) with environmental characteristics such as temperature. Following is an example from `testData/ecoData.csv`:

```
"Site","temp","O2"
"clone",0,50
"m3co",80,40
"m7co",20,30
```

Values in the `Site` column correspond to the `shortSampleName` defined in `sampleDesc.csv`. The file can contain an arbitrary number of environmental variables, and all are used in the constrained ordination inside `ecoAnalysis.R`.

4.5 QIIME mappings file

If QIIME is used (configuration parameter `runQiimeUniFrac="true"`), you must provide a QIIME metadata mapping file, detailed at http://qiime.org/documentation/file_formats.html. FunFrame provides a sample mapping file in `testData/qiime_mapping.txt`.

4.6 Template seed alignment file

If QIIME is used (configuration parameter `runQiimeUniFrac="true"`), you must provide a template alignment for `pynast` [Caporaso et al., 2010a], and set the configuration file parameter `seedAlignmentPath` to this file.

5 Usage

After the configuration files have been setup, the pipeline can be run as a single bash script `run-all-FunFrame`, which runs all of the steps shown in Figure 1.

`run-all-FunFrame` invokes a sequence of `run-*` scripts, which you can run manually in the same sequence. This ordering is necessary to satisfy the dependencies shown in Figure 1:

1. `run-agg-fasta config`
2. `run-hmm-frame config`
3. `run-hmm-frame-score-filter config`
4. `run-uchime-pipeline config`
5. `run-esprit-tree config`
6. `run-make-community-tables config`
7. If QIIME and Unifrac are to be run, invoke the following in this order:
   (a) `run-prep-qiime config`

   (b) run-qiime-alpha-rarefaction config

   (c) run-MSA-FastTree-Unifrac config

8. run-ecoAnalysis config

Invoking run-ecoAnalysis is particularly useful when exploring choices of ecological variables in ecoData.csv, as the upstream files are all unaffected by these choices.

6 Output files

FunFrame produces three types of outputs: log files, data files, and graphical outputs. These files are described below.

6.1 Log files

The underlying run-* scripts produce output log files, and you should check these after a run to ensure that each pipeline stage completed successfully.

- run-agg-fasta.log: This file logs an entry for each input FASTA file, and is useful to check for consistency with sampleDesc.csv.
- run-hmm-frame.log: This is the stderr output of HMMFRAME, which is typically empty.
- run-hmm-frame-score-filter.log: This file shows a histogram of HMMFRAME scores. This information is useful for choosing an appropriate minHMMFRAMEscore in the main configuration file. This file also provides the pre- and post-filtering sequence counts.
- run-uchime-pipeline.log: This file contains the stderr from the runUCHIME.py and uchime.
- run-esprit-tree.log: This file contains the stderr from the ESPRIT-Tree programs preproc, pbpcluster, and invmap.pl.
- run-make-community-tables.log: In addition to general logging of run-make-community-tables and makeOTUtable.R, this file contains a summary of the per-plot OTU counts. For example,

```
OTU counts of m3co:
 sequence_count number_OTUs
              0          16
```

```
   4          2
   5          1
   6          2
   7          1
   9          2
  13          1
  22          1
  36          1
  61          1
  80          1
 130          1
 198          1
1034          1
4044          1
```

indicates that in the `m3co` sample, 16 OTUs were non-represented, and one OTU was represented with 4044 sequence reads.

- The following logs are produced if the configuration parameter `runQiimeUniFrac="true"`. See the underlying run-* scripts and QIIME documentation for full details of the contents.
  - `run-qiime-alpha-rarefaction.log`
  - `run-MSA-FastTree-Unifrac.log`
  - `qiime_mapping.log`
  - `fastTree.log`
- `run-ecoAnalysis.log`: This file contains the result of running `ecoAnalysis.R`. In addition to general logging, this file contains short reports summarizing the sample and OTU counts:

```
[1] "Sequences in each sample"
clone  m3co  m7co
 3992  5668  5078

[1] "OTUs in each sample"
clone  m3co  m7co
    2    17    19
```

This file also contains Vegan/estimateR estimates of community diversity:

```
[1] "Vegan/estimateR"
         clone  m3co  m7co
S.obs     2.00 17.00 19.00
S.chao1   2.00 17.00 19.00
```

However, these estimates are not subsampled as in `run-qiime-alpha-rarefaction`.

6.2 Output files

Most data files contain intermediate results and are not immediately useful outside FunFrame. Several output files, however, are of general use, and are described below.

- `DNA_representative.fasta` and `AA_representative.fasta` are FASTA files containing representative sequences for each OTU. These files are useful for exploring the biological relationships between OTUs and previously characterized sequences, for example, by BLASTing existing microbial sequence databases.
- `reprAnnotatedCommunityTable.csv` is a comma-separated file, readable by spreadsheet programs, that summarizes each OTU with its:
  - Sequence counts in each environmental sample
  - Representative DNA sequence
  - Representative AA sequence
  - FASTA header information with HMMFRAME score.
- `communityTable.csv` is a matrix of a sequence counts, with one column for each OTU and one row for each biological sample. This file is useful for analysis with external tools such as CANOCO.
- `Qiime_communityTable.csv` contains the transposition of `communityTable.csv`, that is, each row represents an OTU and each column is a biological sample. This file is produced regardless of the setting of the configuration parameter `runQiimeUniFrac"`.
- `DNA_representative_aligned.fasta`, produced when `runQiimeUniFrac="true"`, contains the result of the multiple alignment by `pynast`. Inspecting this file is useful to ensure that the subsequent phylogeny is meaningful. This alignment is also useful for identifying conserved and divergent regions of the functional gene being studied.
- `fastTree.out`, produced when `runQiimeUniFrac="true"`, contains an estimated phylogenetic tree in Newick format produced by `FastTree`. This file is useful for additional visualization with external tools such as the Interactive Tree of Life (http://itol.embl.de/).

6.3 Graphical outputs

- `veganAnalysis.pdf`: This file contains plots of rarefaction curves, Shannon diversity with sequence depth, redundancy analysis (RDA), and constrained correspondence analysis (CCA).

- `Bray-Curtis analysis.pdf`: This file contains a heatmap showing the pairwise Bray-Curtis dissimilarity between sites, and a two-dimensional PCoA of the dissimilarity.
- `weighted Unifrac analysis.pdf`: This file is analogous to `Bray-Curtis analysis.pdf` except the weighted UniFrac distance is used.
- `unweighted Unifrac analysis.pdf`: This file is analogous to `Bray-Curtis analysis.pdf` except the unweighted UniFrac distance is used.

References

[Cai and Sun, 2011] Cai, Y. and Sun, Y. (2011). Esprit-tree: hierarchical clustering analysis of millions of 16s rrna pyrosequences in quasilinear computational time. *Nucleic acids research*, 39(14):e95.

[Caporaso et al., 2010a] Caporaso, J. G., Bittinger, K., Bushman, F. D., DeSantis, T. Z., Andersen, G. L., and Knight, R. (2010a). Pynast: a flexible tool for aligning sequences to a template alignment. *Bioinformatics*, 26(2):266–267.

[Caporaso et al., 2010b] Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., Fierer, N., Peña, A. G., Goodrich, J. K., Gordon, J. I., Huttley, G. A., Kelley, S. T., Knights, D., Koenig, J. E., Ley, R. E., Lozupone, C. A., McDonald, D., Muegge, B. D., Pirrung, M., Reeder, J., Sevinsky, J. R., Turnbaugh, P. J., Walters, W. A., Widmann, J., Yatsunenko, T., Zaneveld, J., and Knight, R. (2010b). Qiime allows analysis of high-throughput community sequencing data. *Nature methods*, 7(5):335–336.

[Cock et al., 2009] Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M. J. L. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics (Oxford, England)*, 25(11):1422–1423.

[Edgar et al., 2011] Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., and Knight, R. (2011). Uchime improves sensitivity and speed of chimera detection. *Bioinformatics (Oxford, England)*, 27(16):2194–2200.

[Gihring et al., 2012] Gihring, T. M., Green, S. J., and Schadt, C. W. (2012). Massively parallel rrna gene sequencing exacerbates the potential for biased community diversity comparisons due to variable library sizes. *Environmental microbiology*, 14(2):285–290.

[Hamady et al., 2010] Hamady, M., Lozupone, C., and Knight, R. (2010). Fast unifrac: facilitating high-throughput phylogenetic analyses of microbial communities including analysis of pyrosequencing and phylochip data. *The ISME Journal*, 4(1):17–27.

[Oksanen et al., 2011] Oksanen, J., Blanchet, F. G., Kindt, R., Legendre, P., Minchin, P. R., O'Hara, R. B., Simpson, G. L., Solymos, P., Stevens, M. H. H., and Wagner, H. (2011). *vegan: Community Ecology Package*. R package version 2.0-2.

[Price et al., 2010] Price, M. N., Dehal, P. S., and Arkin, A. P. (2010). Fasttree 2–approximately maximum-likelihood trees for large alignments. *PLoS ONE*, 5(3):e9490.

[R Core Team, 2012] R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[Schloss et al., 2009] Schloss, P. D., Westcott, S. L., Ryabin, T., Hall, J. R., Hartmann, M., Hollister, E. B., Lesniewski, R. A., Oakley, B. B., Parks, D. H., Robinson, C. J., Sahl, J. W., Stres, B., Thallinger, G. G., Horn, D. J. V., and Weber, C. F. (2009). Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and environmental microbiology*, 75(23):7537–7541.

[Zhang and Sun, 2011] Zhang, Y. and Sun, Y. (2011). Hmm-frame: accurate protein domain classification for metagenomic sequences containing frameshift errors. *BMC bioinformatics*, 12:198.