Qiime

- Software is a series of python scripts
- Modular, and incorporates other software and algorithms as dependencies (e.g. BLAST, SWARM, uclust, cd-hit, fastq-join, seqprep, mothur, ChimeraSlayer, uchime, etc.)
  - High number of dependencies makes a native install of qiime….difficult
  - Easy alternatives for installation and use:
    - MacQiime
    - Qiime Virtual box
    - Qiime Amazon Web Services Virtual Machine

Qiime Documentation

- Each script has its own page on the qiime website that lists all of its options

News and Announcements »

Site index

- Home
- Install
- Tutorials
- Scripts
- Help
- Resources
- File Formats
- Workshops
- Blog
- Developer

Previous topic

filter_distance_matrix.py – Filter a distance matrix to contain only a specified set of samples.

Next topic

filter_otus_by_sample.py – Filter OTU mapping file and sequences by SampleIDs

Quick search

[   ] Go

Enter search terms or a module, class or function name.

filter_fasta.py – This script can be applied to remove sequences from a fasta or fastq file based on input criteria.

Description:

Usage: `filter_fasta.py [options]`

Input Arguments:

```
[REQUIRED]

-f, --input_fasta_fp
    Path to the input fasta file

-o, --output_fasta_fp
    The output fasta filepath

[OPTIONAL]

-m, --otu_map
    An OTU map where sequences ids are those which should be retained.

-s, --seq_id_fp
    A list of sequence identifiers (or tab-delimited lines with a seq identifier in the first field) which should
    be retained.

-b, --biom_fp
    A biom file where otu identifiers should be retained.

-a, --subject_fasta_fp
```

Qiime Syntax

- All qiime commands have the same format

Script_name.py --option myvalue –o myvalue

- name of the script you want to call
- usually intuitive eg. join_paired_ends.py, pick_otus.py
- must always be in the first position

Qiime Syntax

- All qiime commands have the same format

Script_name.py --option myvalue –f myvalue

- name of the script you want to call
- usually intuitive eg. join_paired_ends.py, pick_otus.py
- must always be in the first position

Qiime Syntax

- All qiime commands have the same format

Script_name.py --option myvalue –f myvalue
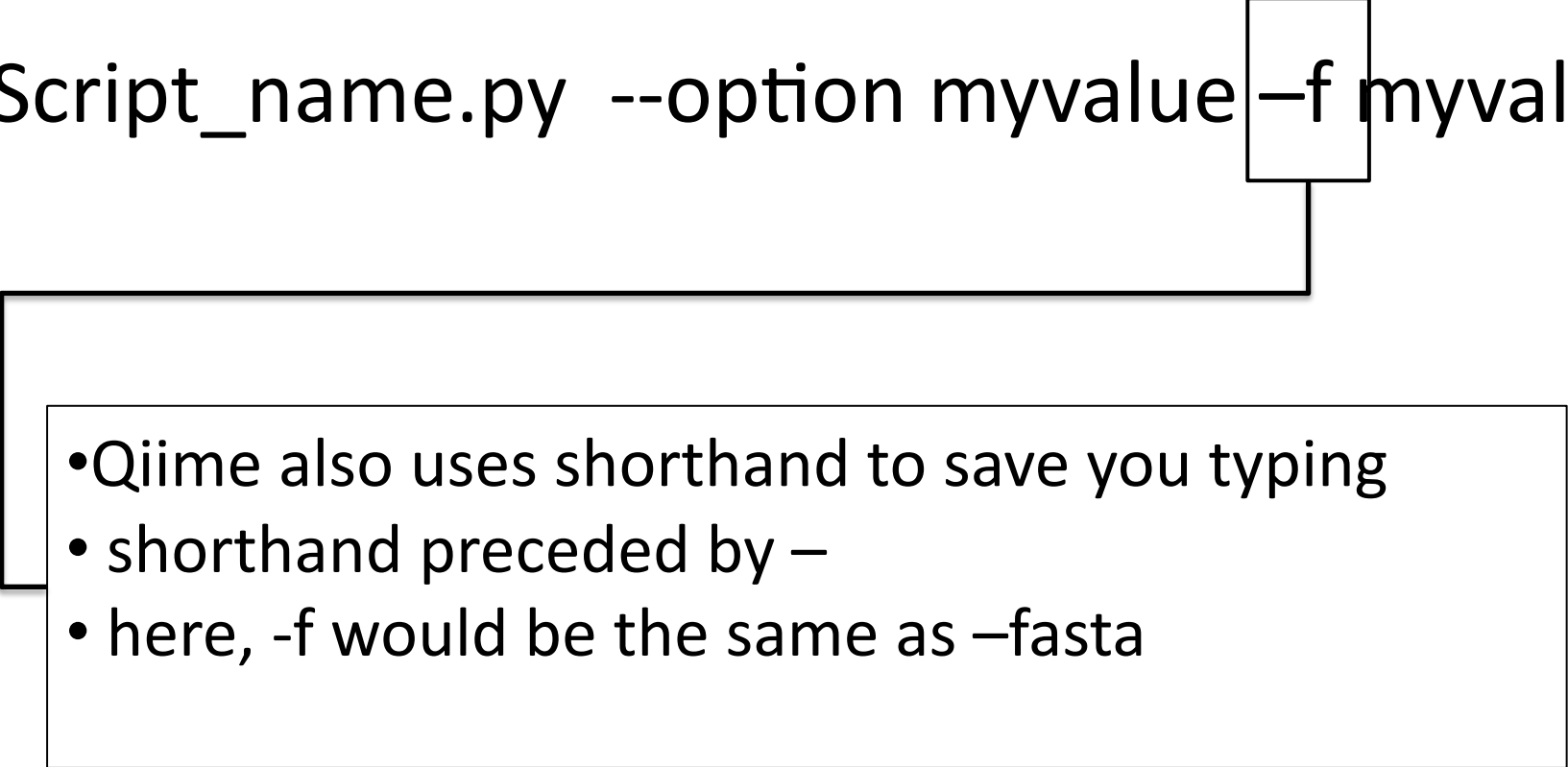
- parameter for the script
- always preceded by --
- usually intuitive eg.
--max_ambig or --fasta
- can have many options in any order

- value for the option
- separated from the option name by a space
- can be a filename, an algorithm name, a numeric value, or a TRUE/FALSE

Qiime Syntax

- All qiime commands have the same format

Script_name.py --option myvalue –f myvalue

- •Qiime also uses shorthand to save you typing
- shorthand preceded by –
- here, -f would be the same as –fasta

Qiime Input Files

1. Sequences! (fastq, fasta + qual, or fasta)
2. Mapping File
   – Specifies metadata about samples
   – Includes list of barcodes, primers, etc.
3. Parameter File
   – Specifies your parameters for multiple
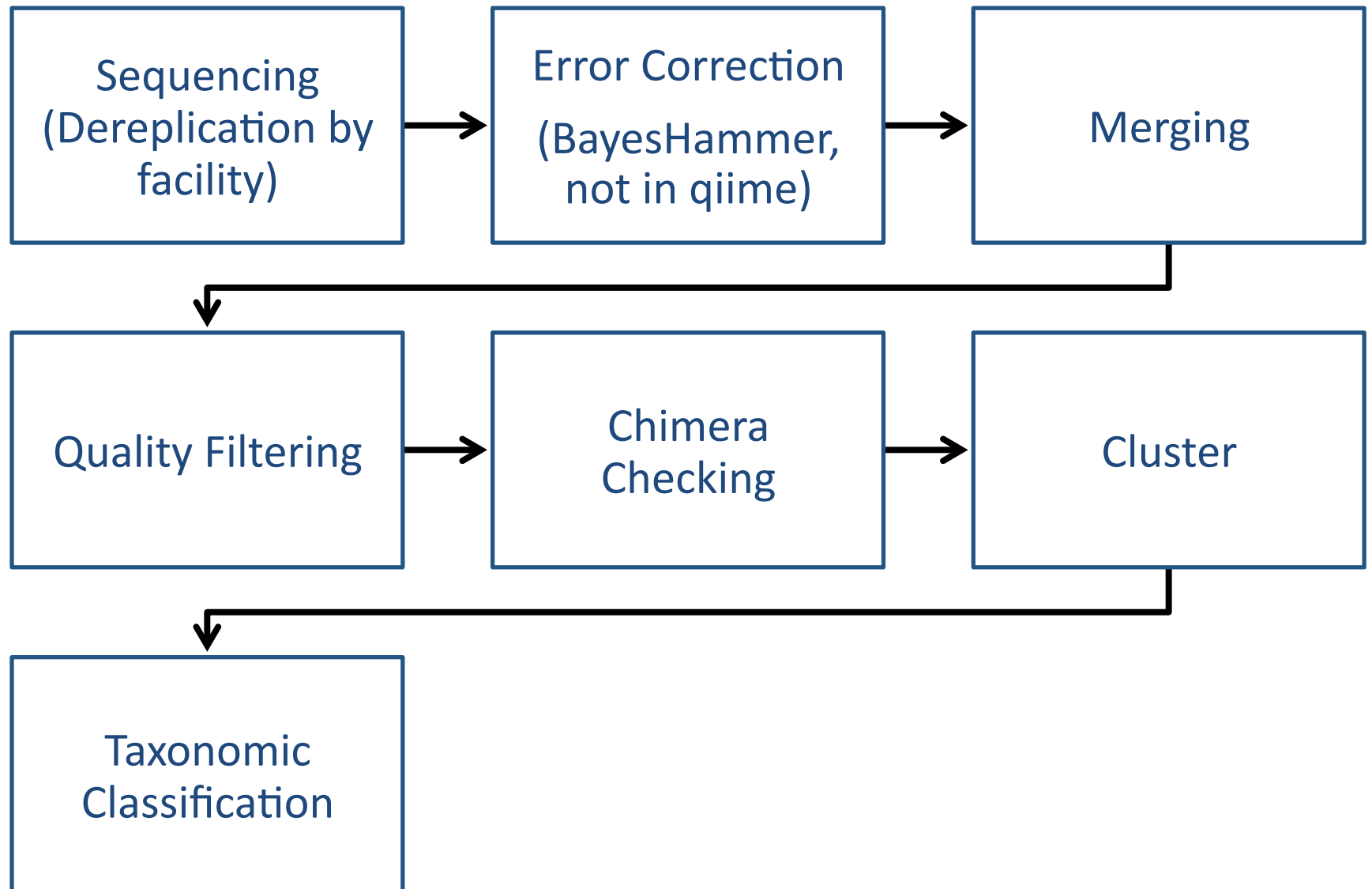
Qiime Input Files: Mapping File

- Tab delimited text file
- Header line is very important
  - First field MUST BE #SampleID, second MUST BE BarcodeSequence, third MUST BE LinkerPrimerSequence, and last MUST BE Description

#SampleID	BarcodeSequence	LinkerPrimerSequence	Treatment	DOB	Description
PC.354	AGCACGAGCCTA	YATGCTGCCTCCCGTAGGAGT	Control	20061218	Control_mouse_I.D._354
PC.355	AACTCGTCGATG	YATGCTGCCTCCCGTAGGAGT	Control	20061218	Control_mouse_I.D._355
PC.356	ACAGACCACTCA	YATGCTGCCTCCCGTAGGAGT	Control	20061126	Control_mouse_I.D._356
PC.481	ACCAGCGACTAG	YATGCTGCCTCCCGTAGGAGT	Control	20070314	Control_mouse_I.D._481
PC.593	AGCAGCACTTGT	YATGCTGCCTCCCGTAGGAGT	Control	20071210	Control_mouse_I.D._593
PC.607	AACTGTGCGTAC	YATGCTGCCTCCCGTAGGAGT	Fast	20071112	Fasting_mouse_I.D._607
PC.634	ACAGAGTCGGCT	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._634
PC.635	ACCGCAGAGTCA	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._635
PC.636	ACGGTGAGTGTC	YATGCTGCCTCCCGTAGGAGT	Fast	20080116	Fasting_mouse_I.D._636

Qiime Input Files: Parameter File

- Parameter files are used for qiime scripts that are 'wrappers'
  - Wrappers run several scripts one after the another, or run the same script multiple times
- Text file where each line specifies a command, option, and value
- Consistent format

  script_name:parameter_name value

```
join_paired_ends:pe_join_method fastq-join
join_paired_ends:min_overlap       10
join_paired_ends:perc_max_diff   90
split_libraries_fastq:max_bad_run_length      3
split_libraries_fastq:min_per_read_length_fraction  0.75
split_libraries_fastq:sequence_max_n      0
split_libraries_fastq:phred_quality_threshold 30
```

```
┌─────────────────────┐      ┌─────────────────────┐      ┌─────────────────────┐
│     Sequencing      │      │  Error Correction   │      │                     │
│ (Dereplication by   │─────▶│                     │─────▶│      Merging        │
│     facility)       │      │  (BayesHammer,      │      │                     │
│                     │      │   not in qiime)     │      │                     │
└─────────────────────┘      └─────────────────────┘      └─────────────────────┘

┌─────────────────────┐      ┌─────────────────────┐      ┌─────────────────────┐
│                     │      │                     │      │                     │
│  Quality Filtering  │─────▶│      Chimera        │─────▶│      Cluster        │
│                     │      │     Checking        │      │                     │
│                     │      │                     │      │                     │
└─────────────────────┘      └─────────────────────┘      └─────────────────────┘

┌─────────────────────┐
│                     │
│     Taxonomic       │
│   Classification    │
│                     │
└─────────────────────┘
```

Qiime Python Scripts Used in our Workflow

- multiple_join_paired_ends.py (merge sequences)
- multiple_split_libraries_fastq.py (quality filter)
- identify_chimeric_seqs.py (chimera check)
- filter_fasta.py (remove chimeras)
- pick_otus.py, pick_open_reference_otus.py (cluster)
- pick_rep_set.py (get representative sequences)
- assign_taxonomy.py (taxonomic assignment)
- make_otu_table.py (build OTU table)
- core_diversity_analyses.py (fun with data!)