# viral-ngs Documentation

***Release v1.23.0***

**Broad Institute Viral Genomics**

**2019-04-24**

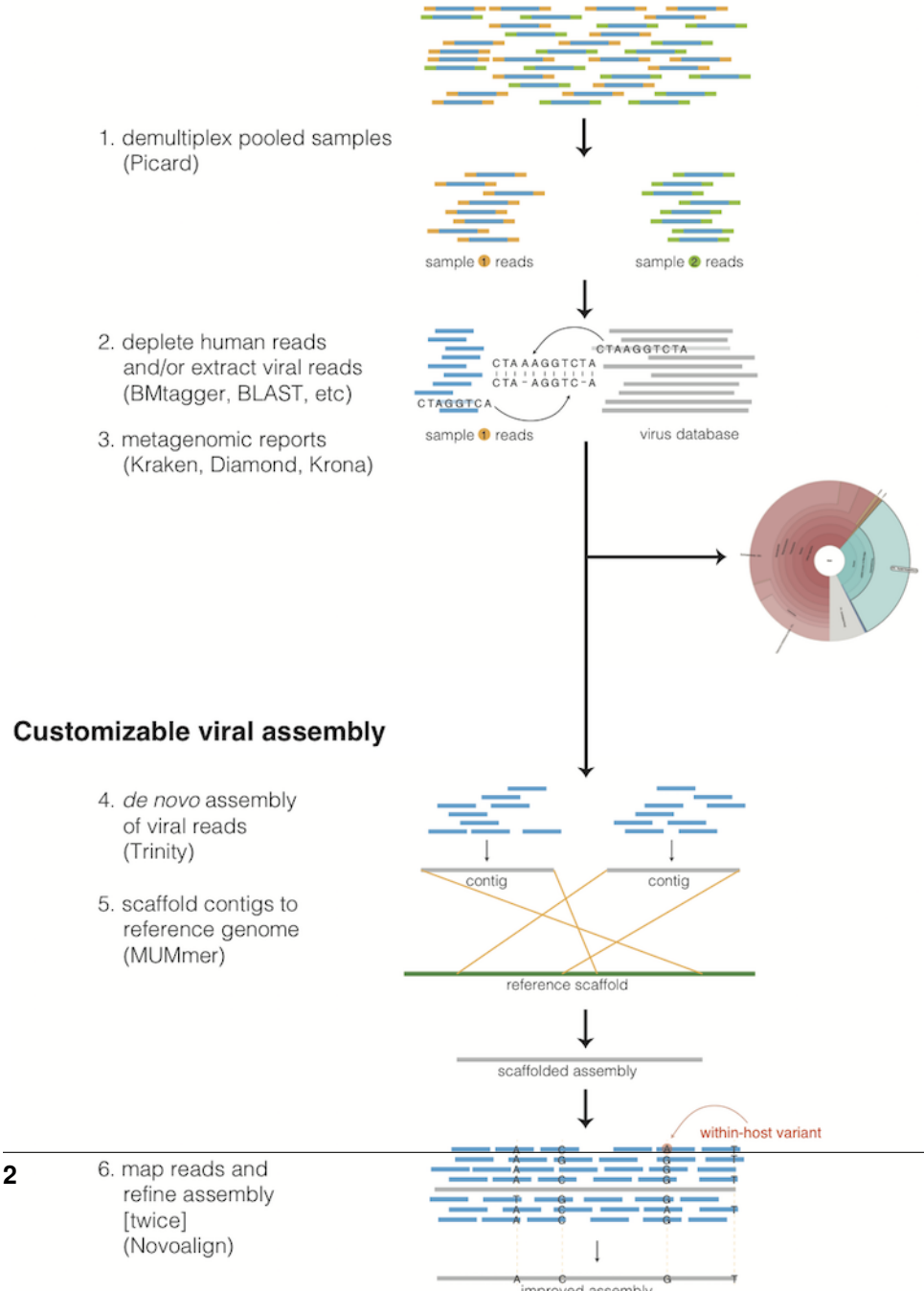
Contents

**1 Contents** **1**
1.1 Description of the methods . . . . . 2
1.2 Installation . . . . . 3
1.3 Command line tools . . . . . 7
1.4 Using the WDL pipelines . . . . . 83
1.5 Using the Snakemake pipelines . . . . . 84

CHAPTER 1

# Contents

## 1.1 Description of the methods



### 1.1.1 Taxonomic read filtration

#### Human, contaminant, and duplicate read removal

The assembly pipeline begins by depleting paired-end reads from each sample of human and other contaminants using BMTAGGER and BLASTN, and removing PCR duplicates using M-Vicuna (a custom version of Vicuna).

#### Taxonomic selection

Reads are then filtered to to a genus-level database using LASTAL, quality-trimmed with Trimmomatic, and further deduplicated with PRINSEQ.

### 1.1.2 Viral genome analysis

#### Viral genome assembly

The filtered and trimmed reads are subsampled to at most 100,000 pairs. *de novo* assemby is performed using Trinity. SPAdes is also offered as an alternative *de novo* assembler. Reference-assisted assembly improvements follow (contig scaffolding, orienting, etc.) with MUMMER and MUSCLE or MAFFT. Gap2Seq is used to seal gaps between scaffolded *de novo* contigs with sequencing reads.

Each sample's reads are aligned to its *de novo* assembly using Novoalign and any remaining duplicates were removed using Picard MarkDuplicates. Variant positions in each assembly were identified using GATK IndelRealigner and UnifiedGenotyper on the read alignments. The assembly was refined to represent the major allele at each variant site, and any positions supported by fewer than three reads were changed to N.

This align-call-refine cycle is iterated twice, to minimize reference bias in the assembly.

#### Intrahost variant identification

Intrahost variants (iSNVs) were called from each sample's read alignments using V-Phaser2 and subjected to an initial set of filters: variant calls with fewer than five forward or reverse reads or more than a 10-fold strand bias were eliminated. iSNVs were also removed if there was more than a five-fold difference between the strand bias of the variant call and the strand bias of the reference call. Variant calls that passed these filters were additionally subjected to a 0.5% frequency filter. The final list of iSNVs contains only variant calls that passed all filters in two separate library preparations. These files infer 100% allele frequencies for all samples at an iSNV position where there was no intra-host variation within the sample, but a clear consensus call during assembly. Annotations are computed with snpEff.

### 1.1.3 Taxonomic read identification

Metagenomic classifiers include Kraken and Diamond. In each case, results are visualized with Krona.

## 1.2 Installation

### 1.2.1 Cloud compute implementations

### Docker Images

To facilitate cloud compute deployments, we publish a complete Docker image with associated dependencies to the Docker registry at quay.io. Simply `docker pull quay.io/broadinstitute/viral-ngs` for the latest stable version.

### DNAnexus

This assembly pipeline is also available via the DNAnexus cloud platform. RNA paired-end reads from either HiSeq or MiSeq instruments can be securely uploaded in FASTQ or BAM format and processed through the pipeline using graphical and command-line interfaces. Instructions for the cloud analysis pipeline are available at https://github.com/dnanexus/viral-ngs/wiki

### Google Cloud Platform: deploy to GCE VM

The docker image referenced above can be directly deployed to a Google Compute Engine VM on startup. The main things you will need to do are:

- Make sure to allocate a larger-than-default root disk for the VM. Google's Container Optimized OS defaults to a very small disk which is not large enough to unpack our Docker image. Increase to at least 20GB (or more if you want to localize data).
- When setting up the VM for launch, make sure you open the "Advanced container options" hidden options and select "Allocate a buffer for STDIN" and "Allocate a pseudo-TTY" before launching. Otherwise you won't be able to ssh into them!
- Sometimes you will need to invoke "bash" manually upon login to get the correct environment.

### Google Cloud Platform: dsub

All of the command line functions in viral-ngs are accessible from the docker image and can be invoked directly using dsub.

Here is an example invocation of `illumina.py illumina_demux` (replace the project with your GCP project, and the input, output-recursive, and logging parameters with URIs within your GCS buckets):

```
dsub --project gcid-viral-seq --zones "us-central1-*" \
  --image quay.io/broadinstitute/viral-ngs \
  --name illumina_demux-test \
  --logging gs://viral-temp-30d/dpark/test-demux/logs \
  --input FC_TGZ=gs://viral-sequencing/flowcells/broad-walkup/160907_M04004_0066_
↪000000000-AJH8U.tar.gz \
  --output-recursive OUTDIR=gs://viral-temp-30d/dpark/test-demux \
  --command 'illumina.py illumina_demux ${FC_TGZ} 1 ${OUTDIR}' \
  --min-ram 30 \
  --min-cores 8 \
  --disk-size 2000
```

The speed of disk write and read operations is linearly proportional to the disk size, hitting the maximum disk speed somewhere around 1-8TB (depending on your I/O pattern). See GCE documentation.

## 1.2.2 Manual Installation

### Install Conda

To use viral-ngs, you need to install the Conda package manager which is most easily obtained via the Miniconda Python distribution. Miniconda can be installed on your system without admin priviledges.

After installing Miniconda for your platform, be sure to update it:

```
conda update -y conda
```

### Configure Conda

The viral-ngs software and its dependencies are distributed through the a channel of the conda package manager. It is necessary to add this channel to the conda config:

```
conda config --add channels r
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
conda config --add channels broad-viral
```

### Make a conda environment and install viral-ngs

It is recommended to install viral-ngs into its own conda environment. This ensures its dependencies do not interfere with other conda packages installed on your system. A new conda environment can be created with the following command, which will also install conda:

```
conda create -n viral-ngs-env viral-ngs
```

### Activate the viral-ngs environment and complete the install

In order to finish installing viral-ngs, you will need to activate its conda environment:

```
source activate viral-ngs-env
```

Due to license restrictions, the viral-ngs conda package cannot distribute and install GATK directly. To fully install GATK, you must download a licensed copy of GATK v3.6 from the Broad Institute, and call "gatk3-register," which will copy GATK into your viral-ngs conda environment:

```
mkdir -p /path/to/gatk_dir
wget -O - 'https://software.broadinstitute.org/gatk/download/auth?package=GATK-
↪archive&version=3.6-0-g89b7209' | tar -xjvC /path/to/gatk_dir
gatk3-register /path/to/gatk_dir/GenomeAnalysisTK.jar
```

The single-threaded version of Novoalign is installed by default. If you have a license for Novoalign to enable multi-threaded operation, viral-ngs will copy it to the viral-ngs conda environment if the `NOVOALIGN_LICENSE_PATH` environment variable is set. Alternatively, the conda version of Novoalign can be overridden if the `NOVOALIGN_PATH` environment variable is set. If you obtain a Novoalign license after viral-ngs has already been installed, it can be added to the conda environment by calling:

```
# obtain a Novoalign license file: novoalign.lic
novoalign-license-register /path/to/novoalign.lic
```

#### Activating viral-ngs once installed

After viral-ngs has been installed, only one command is needed to load the environment and all of its dependencies. This is the command that must be run each time before using viral-ngs:

```
source activate viral-ngs-env
```

To deactivate the conda environment:

```
source deactivate
```

### 1.2.3 Easy deployment script for viral-ngs

**viral-ngs** can be deployed with help of a shell script, `easy-deploy/easy-deploy-viral-ngs.sh` . This script will install an independent copy of viral-ngs from the latest source, install all dependencies, and make it simple to activate the viral-ngs environment and create projects. The script is available from the repository broadinstitute/viral-ngs.

After downloading the script and making it executable, viral-ngs can be installed on a 64-bit macOS or Linux system via:

```
./easy-deploy-viral-ngs.sh setup
```

#### One-line install command

This one-line command will install viral-ngs on a 64-bit macOS or Linux system:

```
wget https://raw.githubusercontent.com/broadinstitute/viral-ngs/master/easy-deploy-
↪script/easy-deploy-viral-ngs.sh && chmod a+x ./easy-deploy-viral-ngs.sh && ./easy-
↪deploy-viral-ngs.sh setup
```

#### One-line install command for Broad Institute cluster users

This one-line command will download the `easy-deploy-viral-ngs.sh` script and setup viral-ngs in the current working directory. Simply ssh to one of the Broad login nodes and paste this command:

```
wget https://raw.githubusercontent.com/broadinstitute/viral-ngs/master/easy-deploy-
↪script/easy-deploy-viral-ngs.sh && chmod a+x ./easy-deploy-viral-ngs.sh && reuse␣
↪UGER && qrsh -l h_vmem=10G -cwd -N "viral-ngs_deploy" -q interactive ./easy-deploy-
↪viral-ngs.sh setup
```

**Note:** The script will run the install on a UGER interactive node, so you must have the ability to create to start a new interactive session. A project can be specified via `qrsh -P "<project_name>"`

#### Usage

**Installation**

- `./easy-deploy-viral-ngs.sh setup` Installs a fresh copy of viral-ngs, installs all dependencies, and creates a directory, `viral-ngs-etc/` , in the current working directory.

Resulting directories:

```
viral-ngs-etc/
    conda-env/
    viral-ngs/
    mc3/
```

**Activating the environment**

- `source ./easy-deploy-viral-ngs.sh load` Loads the dotkits needed by viral-ngs and activates the Python virtual environment

**Creating a project directory**

- `./easy-deploy-viral-ngs.sh create-project <project_name>` Creates a directory for a new Snakemake-compatible project, with data directories and symlinked run scripts. Copies in the files `Snakefile` and `config.yaml`

Resulting directories:

```
viral-ngs-analysis-software/
    projects/
        <project_name>/
            Snakefile
            bin/ (symlink)
            config.yaml
            data/
            log/
            reports/
            run-pipe_LSF.sh (symlink)
            run-pipe_UGER.sh (symlink)
            samples-assembly-failures.txt
            samples-assembly.txt
            samples-depletion.txt
            samples-runs.txt
            tmp/
            venv/ (symlink)
            [...other project files...]
```

## 1.3 Command line tools

### 1.3.1 metagenomics.py - metagenomic analyses

This script contains a number of utilities for metagenomic analyses.

```
usage: metagenomics.py subcommand
```

**Sub-commands:**

**subset_taxonomy**

Generate a subset of the taxonomy db files filtered by the whitelist. The whitelist taxids indicate specific taxids plus their parents to add to taxonomy while whitelistTreeTaxids indicate specific taxids plus both parents and all children taxa. Whitelist GI and accessions can only be provided in file form and the resulting gi/accession2taxid files will be filtered to only include those in the whitelist files. Finally, taxids + parents for the gis/accessions will also be included.

```
usage: metagenomics.py subset_taxonomy [-h]
                                       [--whitelistTaxids WHITELISTTAXIDS␣
↪[WHITELISTTAXIDS ...]]
                                       [--whitelistTaxidFile␣
↪WHITELISTTAXIDFILE]
                                       [--whitelistTreeTaxids␣
↪WHITELISTTREETAXIDS [WHITELISTTREETAXIDS ...]]
                                       [--whitelistTreeTaxidFile␣
↪WHITELISTTREETAXIDFILE]
                                       [--whitelistGiFile WHITELISTGIFILE]
                                       [--whitelistAccessionFile␣
↪WHITELISTACCESSIONFILE]
                                       [--skipGi] [--skipAccession]
                                       [--skipDeadAccession]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version] [--tmp_dir TMP_DIR]
                                       [--tmp_dirKeep]
                                      taxDb outputDb
```

**Positional arguments:**

**taxDb** Taxonomy database directory (containing nodes.dmp, parents.dmp etc.)

**outputDb** Output taxonomy database directory

**Options:**

**--whitelistTaxids** List of taxids to add to taxonomy (with parents)

**--whitelistTaxidFile** File containing taxids - one per line - to add to taxonomy with parents.

**--whitelistTreeTaxids** List of taxids to add to taxonomy (with parents and children)

**--whitelistTreeTaxidFile** File containing taxids - one per line - to add to taxonomy with parents and children.

**--whitelistGiFile** File containing GIs - one per line - to add to taxonomy with nodes.

**--whitelistAccessionFile** File containing accessions - one per line - to add to taxonomy with nodes.

**--skipGi=False** Skip GI to taxid mapping files

**--skipAccession=False** Skip accession to taxid mapping files

**--skipDeadAccession=False** Skip dead accession to taxid mapping files

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**krakenuniq**

Classify reads by taxon using KrakenUniq

```
usage: metagenomics.py krakenuniq [-h]
                                  [--outReports OUTREPORTS [OUTREPORTS ...]]
                                  [--outReads OUTREADS [OUTREADS ...]]
                                  [--filterThreshold FILTERTHRESHOLD]
                                  [--threads THREADS]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep]
                                  db inBams [inBams ...]
```

**Positional arguments:**

| | |
|---|---|
| **db** | Kraken database directory. |
| **inBams** | Input unaligned reads, BAM format. |

**Options:**

| | |
|---|---|
| **--outReports** | Kraken summary report output file. Multiple filenames space separated. |
| **--outReads** | Kraken per read classification output file. Multiple filenames space separated. |
| **--filterThreshold=0.05** | Kraken filter threshold (default %(default)s) |
| **--threads** | Number of threads (default: all available cores) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**krona**

Create an interactive HTML report from a tabular metagenomic report

```
usage: metagenomics.py krona [-h] [--queryColumn QUERYCOLUMN]
                             [--taxidColumn TAXIDCOLUMN]
                             [--scoreColumn SCORECOLUMN]
                             [--magnitudeColumn MAGNITUDECOLUMN] [--noHits]
                             [--noRank] [--inputType {tsv,krakenuniq,kaiju}]
                             [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                             [--version]
                             inReport db outHtml
```

**Positional arguments:**

| | |
|---|---|
| **inReport** | Input report file (default: tsv) |
| **db** | Krona taxonomy database directory. |

| | |
|---|---|
| **outHtml** | Output html report. |

**Options:**

| | |
|---|---|
| **--queryColumn=2** | Column of query id. (default %(default)s) |
| **--taxidColumn=3** | Column of taxonomy id. (default %(default)s) |
| **--scoreColumn** | Column of score. (default %(default)s) |
| **--magnitudeColumn** | Column of magnitude. (default %(default)s) |
| **--noHits=False** | Include wedge for no hits. |
| **--noRank=False** | Include no rank assignments. |
| **--inputType=tsv** | Handling for specialized report types.<br>Possible choices: tsv, krakenuniq, kaiju |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**kaiju**

Classify reads by the taxon of the Lowest Common Ancestor (LCA)

```
usage: metagenomics.py kaiju [-h] [--outReads OUTREADS] [--threads THREADS]
                             [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                             [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                             inBam db taxDb outReport
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input unaligned reads, BAM format. |
| **db** | Kaiju database .fmi file. |
| **taxDb** | Taxonomy database directory. |
| **outReport** | Output taxonomy report. |

**Options:**

| | |
|---|---|
| **--outReads** | Output LCA assignments for each read. |
| **--threads** | Number of threads (default: all available cores) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**report_merge**

Merge multiple metegenomic reports into a single metagenomic report. Any Krona input files created by this

```
usage: metagenomics.py report_merge [-h]
                                    [--outSummaryReport OUT_KRAKEN_SUMMARY]
                                    [--krakenDB KRAKEN_DB]
                                    [--outByQueryToTaxonID OUT_KRONA_INPUT]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    metagenomic_reports
                                    [metagenomic_reports ...]
```

**Positional arguments:**

**metagenomic_reports** Input metagenomic reports with the query ID and taxon ID in the 2nd and 3rd columns (Kraken format)

**Options:**

**--outSummaryReport** Path of human-readable metagenomic summary report, created by kraken-report

**--krakenDB** Kraken database (needed for outSummaryReport)

**--outByQueryToTaxonID** Output metagenomic report suitable for Krona input.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**filter_bam_to_taxa**

Filter an (already classified) input bam file to only include reads that have been mapped to specified taxonomic IDs or scientific names. This requires a classification file, as produced by tools such as Kraken, as well as the NCBI taxonomy database.

```
usage: metagenomics.py filter_bam_to_taxa [-h]
                                          [--taxNames TAX_NAMES [TAX_NAMES ...
↪]]
                                          [--taxIDs TAX_IDS [TAX_IDS ...]]
                                          [--without-children]
                                          [--read_id_col READ_ID_COL]
                                          [--tax_id_col TAX_ID_COL]
                                          [--JVMmemory JVMMEMORY]
                                          [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                          [--version] [--tmp_dir TMP_DIR]
                                          [--tmp_dirKeep]
                                          in_bam read_IDs_to_tax_IDs out_bam
                                          nodes_dmp names_dmp
```

**Positional arguments:**

**in_bam** Input bam file.

**read_IDs_to_tax_IDs** TSV file mapping read IDs to taxIDs, Kraken-format by default. Assumes bijective mapping of read ID to tax ID.

**out_bam** Output bam file, filtered to the taxa specified

**nodes_dmp** nodes.dmp file from ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/

**names_dmp** names.dmp file from ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/

**Options:**

**--taxNames** The taxonomic names to include. More than one can be specified. Mapped to Tax IDs by lowercase exact match only. Ex. "Viruses" This is in addition to any taxonomic IDs provided.

**--taxIDs** The NCBI taxonomy IDs to include. More than one can be specified. This is in addition to any taxonomic names provided.

**--without-children=False** Omit reads classified more specifically than each taxon specified (without this a taxon and its children are included).

**--read_id_col=1** The (zero-indexed) number of the column in read_IDs_to_tax_IDs containing read IDs. (default: %(default)s)

**--tax_id_col=2** The (zero-indexed) number of the column in read_IDs_to_tax_IDs containing Taxonomy IDs. (default: %(default)s)

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**taxlevel_summary**

Aggregates taxonomic abundance data from multiple Kraken-format summary files. It is intended to report information on a particular taxonomic level (–taxlevelFocus; ex. 'species'), within a higher-level grouping (–taxHeading; ex. 'Viruses'). By default, when –taxHeading is at the same level as –taxlevelFocus a summary with lines for each sample is emitted. Otherwise, a histogram is returned. If per-sample information is desired, –noHist can be specified. In per-sample data, the suffix "-pt" indicates percentage, so a value of 0.02 is 0.0002 of the total number of reads for the sample. If –topN is specified, only the top N most abundant taxa are included in the histogram count or per-sample output. If a number is specified for –countThreshold, only taxa with that number of reads (or greater) are included. Full data returned via –jsonOut (filtered by –topN and –countThreshold), whereas -csvOut returns a summary.

```
usage: metagenomics.py taxlevel_summary [-h] [--jsonOut JSON_OUT]
                                        [--csvOut CSV_OUT]
                                        [--taxHeading TAX_HEADINGS [TAX_
↪HEADINGS ...]]
                                        [--taxlevelFocus
↪{species,genus,family,order,class,phylum,kingdom,superkingdom}]
                                        [--topN TOP_N_ENTRIES]
                                        [--countThreshold COUNT_THRESHOLD]
```

```
                                        [--zeroFill] [--noHist]
                                        [--includeRoot]
                                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                        [--version] [--tmp_dir TMP_DIR]
                                        [--tmp_dirKeep]
                                        summary_files_in
                                        [summary_files_in ...]
```

**Positional arguments:**

**summary_files_in** Kraken-format summary text file with tab-delimited taxonomic levels.

**Options:**

**--jsonOut** The path to a json file containing the relevant parsed summary data in json format.

**--csvOut** The path to a csv file containing sample-specific counts.

**--taxHeading=Viruses** The taxonomic heading to analyze (default: %(default)s). More than one can be specified.

**--taxlevelFocus=species** The taxonomic heading to summarize (totals by Genus, etc.) (default: %(default)s).

Possible choices: species, genus, family, order, class, phylum, kingdom, superkingdom

**--topN=100** Only include the top N most abundant taxa by read count (default: %(default)s)

**--countThreshold=1** Minimum number of reads to be included (default: %(default)s)

**--zeroFill=False** When absent from a sample, write zeroes (rather than leaving blank).

**--noHist=False** Write out a report by-sample rather than a histogram.

**--includeRoot=False** Include the count of reads at the root level and the unclassified bin.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**krakenuniq_build**

Builds a krakenuniq database from library directory of fastas and taxonomy db directory. The –subsetTaxonomy option allows shrinking the taxonomy to only include taxids associated with the library folders. For this to work, the library fastas must have the standard accession id names such as '>NC1234.1' or '>NC_01234.1'.

Setting the –minimizerLen (default: 16) small, such as 10, will drastically shrink the db size for small inputs, which is useful for testing.

The built db may include symlinks to the original –library / –taxonomy directories. If you want to build a static archiveable version of the library, simply use the –clean option, which will also remove any unnecessary files.

```
usage: metagenomics.py krakenuniq_build [-h] [--library LIBRARY]
                                        [--taxonomy TAXONOMY]
                                        [--subsetTaxonomy]
                                        [--minimizerLen MINIMIZERLEN]
                                        [--kmerLen KMERLEN]
                                        [--maxDbSize MAXDBSIZE] [--clean]
                                        [--workOnDisk] [--threads THREADS]
                                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                        [--version] [--tmp_dir TMP_DIR]
                                        [--tmp_dirKeep]
                                        db
```

**Positional arguments:**

| | |
|---|---|
| **db** | Krakenuniq database output directory. |

**Options:**

| | |
|---|---|
| **--library** | Input library directory of fasta files. If not specified, it will be read from the "library" subdirectory of "db". |
| **--taxonomy** | Taxonomy db directory. If not specified, it will be read from the "taxonomy" subdirectory of "db". |
| **--subsetTaxonomy=False** | Subset taxonomy based on library fastas. |
| **--minimizerLen** | Minimizer length (krakenuniq default: 15) |
| **--kmerLen** | k-mer length (krakenuniq default: 31) |
| **--maxDbSize** | Maximum db size in GB (will shrink if too big) |
| **--clean=False** | Clean by deleting other database files after build |
| **--workOnDisk=False** | Work on disk instead of RAM. This is generally much slower unless the "db" directory lives on a RAM disk. |
| **--threads** | Number of threads (default: all available cores) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

### 1.3.2 taxon_filter.py - tools for taxonomic removal or filtration of reads

This script contains a number of utilities for filtering NGS reads based on membership or non-membership in a species / genus / taxonomic grouping.

```
usage: taxon_filter.py subcommand
```

**Sub-commands:**

**deplete**

Run the entire depletion pipeline: bwa, bmtagger, mvicuna, blastn.

```
usage: taxon_filter.py deplete [-h] [--bwaDbs [BWADBS [BWADBS ...]]]
                               [--bmtaggerDbs [BMTAGGERDBS [BMTAGGERDBS ...]]]
                               [--blastDbs [BLASTDBS [BLASTDBS ...]]]
                               [--srprismMemory SRPRISM_MEMORY]
                               [--chunkSize CHUNKSIZE] [--JVMmemory JVMMEMORY]
                               [--clearTags]
                               [--tagsToClear TAGS_TO_CLEAR [TAGS_TO_CLEAR ...
↪]]
                               [--doNotSanitize] [--threads THREADS]
                               [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                               [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                               inBam [revertBam] bwaBam bmtaggerBam rmdupBam
                               blastnBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input BAM file. |
| **revertBam** | Output BAM: read markup reverted with Picard. |
| **bwaBam** | Output BAM: depleted of reads with BWA. |
| **bmtaggerBam** | Output BAM: depleted of reads with BMTagger. |
| **rmdupBam** | Output BAM: bmtaggerBam run through M-Vicuna duplicate removal. |
| **blastnBam** | Output BAM: rmdupBam run through another depletion of reads with BLASTN. |

**Options:**

| | |
|---|---|
| **--bwaDbs=()** | Reference databases for blast to deplete from input. |
| **--bmtaggerDbs=()** | Reference databases to deplete from input. For each db, requires prior creation of db.bitmask by bmtool, and db.srprism.idx, db.srprism.map, etc. by srprism mkindex. |
| **--blastDbs=()** | Reference databases for blast to deplete from input. |
| **--srprismMemory=7168** | Memory for srprism. |
| **--chunkSize=1000000** | blastn chunk size (default: %(default)s) |
| **--JVMmemory=4g** | JVM virtual memory size for Picard FilterSamReads (default: %(default)s) |
| **--clearTags=False** | When supplying an aligned input file, clear the per-read attribute tags |
| **--tagsToClear=['XT', 'X0', 'X1', 'XA', 'AM', 'SM', 'BQ', 'CT', 'XN', 'OC', 'OP']** | A space-separated list of tags to remove from all reads in the input bam file (default: %(default)s) |
| **--doNotSanitize=False** | Undocumented |

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**deplete_human**

A wrapper around 'deplete'; deprecated but preserved for legacy compatibility.

```
usage: taxon_filter.py deplete_human [-h] [--bwaDbs [BWADBS [BWADBS ...]]]
                                     [--bmtaggerDbs [BMTAGGERDBS [BMTAGGERDBS␣
↪...]]]
                                     [--blastDbs [BLASTDBS [BLASTDBS ...]]]
                                     [--srprismMemory SRPRISM_MEMORY]
                                     [--chunkSize CHUNKSIZE]
                                     [--JVMmemory JVMMEMORY] [--clearTags]
                                     [--tagsToClear TAGS_TO_CLEAR [TAGS_TO_
↪CLEAR ...]]
                                     [--doNotSanitize] [--threads THREADS]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version] [--tmp_dir TMP_DIR]
                                     [--tmp_dirKeep]
                                     inBam [revertBam] bwaBam bmtaggerBam
                                     rmdupBam blastnBam
```

**Positional arguments:**

**inBam** Input BAM file.

**revertBam** Output BAM: read markup reverted with Picard.

**bwaBam** Output BAM: depleted of reads with BWA.

**bmtaggerBam** Output BAM: depleted of reads with BMTagger.

**rmdupBam** Output BAM: bmtaggerBam run through M-Vicuna duplicate removal.

**blastnBam** Output BAM: rmdupBam run through another depletion of reads with BLASTN.

**Options:**

**--bwaDbs=()** Reference databases for blast to deplete from input.

**--bmtaggerDbs=()** Reference databases to deplete from input. For each db, requires prior creation of db.bitmask by bmtool, and db.srprism.idx, db.srprism.map, etc. by srprism mkindex.

**--blastDbs=()** Reference databases for blast to deplete from input.

**--srprismMemory=7168** Memory for srprism.

**--chunkSize=1000000** blastn chunk size (default: %(default)s)

**--JVMmemory=4g** JVM virtual memory size for Picard FilterSamReads (default: %(default)s)

**--clearTags=False** When supplying an aligned input file, clear the per-read attribute tags

**--tagsToClear=['XT', 'X0', 'X1', 'XA', 'AM', 'SM', 'BQ', 'CT', 'XN', 'OC', 'OP']** A space-separated list of tags to remove from all reads in the input bam file (default: %(default)s)

**--doNotSanitize=False** Undocumented

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**filter_lastal_bam**

Restrict input reads to those that align to the given reference database using LASTAL.

```
usage: taxon_filter.py filter_lastal_bam [-h]
                                         [-n MAX_GAPLESS_ALIGNMENTS_PER_
↪POSITION]
                                         [-l MIN_LENGTH_FOR_INITIAL_MATCHES]
                                         [-L MAX_LENGTH_FOR_INITIAL_MATCHES]
                                         [-m MAX_INITIAL_MATCHES_PER_POSITION]
                                         [--JVMmemory JVMMEMORY]
                                         [--threads THREADS]
                                         [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                         [--version] [--tmp_dir TMP_DIR]
                                         [--tmp_dirKeep]
                                         inBam db outBam
```

**Positional arguments:**

**inBam** Input reads

**db** Database of taxa we keep

**outBam** Output reads, filtered to refDb

**Options:**

**-n=1** maximum gapless alignments per query position (default: %(default)s)

**-l=5** minimum length for initial matches (default: %(default)s)

**-L=50** maximum length for initial matches (default: %(default)s)

**-m=100** maximum initial matches per query position (default: %(default)s)

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**deplete_bam_bmtagger**

Use bmtagger to deplete input reads against several databases.

```
usage: taxon_filter.py deplete_bam_bmtagger [-h]
                                            [--srprismMemory SRPRISM_MEMORY]
                                            [--JVMmemory JVMMEMORY]
                                            [--clearTags]
                                            [--tagsToClear TAGS_TO_CLEAR␣
↪[TAGS_TO_CLEAR ...]]
                                            [--doNotSanitize]
                                            [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                            [--version] [--tmp_dir TMP_DIR]
                                            [--tmp_dirKeep]
                                            inBam refDbs [refDbs ...] outBam
```

**Positional arguments:**

**inBam** Input BAM file.

**refDbs** Reference databases (one or more) to deplete from input. For each db, requires prior creation of db.bitmask by bmtool, and db.srprism.idx, db.srprism.map, etc. by srprism mkindex.

**outBam** Output BAM file.

**Options:**

**--srprismMemory=7168** Memory for srprism.

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--clearTags=False** When supplying an aligned input file, clear the per-read attribute tags

**--tagsToClear=['XT', 'X0', 'X1', 'XA', 'AM', 'SM', 'BQ', 'CT', 'XN', 'OC', 'OP']** A space-separated list of tags to remove from all reads in the input bam file (default: %(default)s)

**--doNotSanitize=False** Undocumented

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**deplete_blastn_bam**

Use blastn to remove reads that match at least one of the specified databases.

```
usage: taxon_filter.py deplete_blastn_bam [-h] [--chunkSize CHUNKSIZE]
                                          [--JVMmemory JVMMEMORY]
                                          [--clearTags]
                                          [--tagsToClear TAGS_TO_CLEAR [TAGS_
↪TO_CLEAR ...]]
                                          [--doNotSanitize]
                                          [--threads THREADS]
                                          [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                          [--version] [--tmp_dir TMP_DIR]
                                          [--tmp_dirKeep]
                                          inBam refDbs [refDbs ...] outBam
```

**Positional arguments:**

**inBam** Input BAM file.

**refDbs** One or more reference databases for blast. An ephemeral database will be created if a fasta file is provided.

**outBam** Output BAM file with matching reads removed.

**Options:**

**--chunkSize=1000000** FASTA chunk size (default: %(default)s)

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--clearTags=False** When supplying an aligned input file, clear the per-read attribute tags

**--tagsToClear=['XT', 'X0', 'X1', 'XA', 'AM', 'SM', 'BQ', 'CT', 'XN', 'OC', 'OP']** A space-separated list of tags to remove from all reads in the input bam file (default: %(default)s)

**--doNotSanitize=False** Undocumented

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**deplete_bwa_bam**

Use BWA to remove reads that match at least one of the specified databases.

```
usage: taxon_filter.py deplete_bwa_bam [-h] [--clearTags]
                                       [--tagsToClear TAGS_TO_CLEAR [TAGS_TO_
↪CLEAR ...]]
```

```
                                        [--doNotSanitize]
                                        [--JVMmemory JVMMEMORY]
                                        [--threads THREADS]
                                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                        [--version] [--tmp_dir TMP_DIR]
                                        [--tmp_dirKeep]
                                        inBam refDbs [refDbs ...] outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input BAM file. |
| **refDbs** | One or more reference databases for bwa. An ephemeral database will be created if a fasta file is provided. |
| **outBam** | Ouput BAM file with matching reads removed. |

**Options:**

| | |
|---|---|
| **--clearTags=False** | When supplying an aligned input file, clear the per-read attribute tags |
| **--tagsToClear=['XT', 'X0', 'X1', 'XA', 'AM', 'SM', 'BQ', 'CT', 'XN', 'OC', 'OP']** | A space-separated list of tags to remove from all reads in the input bam file (default: %(default)s) |
| **--doNotSanitize=False** | Undocumented |
| **--JVMmemory=2g** | JVM virtual memory size (default: %(default)s) |
| **--threads** | Number of threads (default: all available cores) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**lastal_build_db**

build a database for use with last based on an input fasta file

```
usage: taxon_filter.py lastal_build_db [-h]
                                        [--outputFilePrefix OUTPUTFILEPREFIX]
                                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                        [--version] [--tmp_dir TMP_DIR]
                                        [--tmp_dirKeep]
                                        inputFasta outputDirectory
```

**Positional arguments:**

| | |
|---|---|
| **inputFasta** | Location of the input FASTA file |
| **outputDirectory** | Location for the output files (default is cwd: %(default)s) |

**Options:**

**--outputFilePrefix** Prefix for the output file name (default: inputFasta name, sans ".fasta" extension)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**bwa_build_db**

Create a database for use with bwa from an input reference FASTA file

```
usage: taxon_filter.py bwa_build_db [-h] [--outputFilePrefix OUTPUTFILEPREFIX]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    inputFasta outputDirectory
```

**Positional arguments:**

**inputFasta** Location of the input FASTA file

**outputDirectory** Location for the output files

**Options:**

**--outputFilePrefix** Prefix for the output file name (default: inputFasta name, sans ".fasta" extension)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**blastn_build_db**

Create a database for use with blastn from an input reference FASTA file

```
usage: taxon_filter.py blastn_build_db [-h]
                                       [--outputFilePrefix OUTPUTFILEPREFIX]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version] [--tmp_dir TMP_DIR]
                                       [--tmp_dirKeep]
                                       inputFasta outputDirectory
```

**Positional arguments:**

**inputFasta** Location of the input FASTA file

**outputDirectory** Location for the output files

**Options:**

**--outputFilePrefix** Prefix for the output file name (default: inputFasta name, sans ".fasta" extension)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**bmtagger_build_db**

Create a database for use with Bmtagger from an input FASTA file.

```
usage: taxon_filter.py bmtagger_build_db [-h]
                                         [--outputFilePrefix OUTPUTFILEPREFIX]
                                         [--word_size WORD_SIZE]
                                         [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                         [--version] [--tmp_dir TMP_DIR]
                                         [--tmp_dirKeep]
                                         inputFasta outputDirectory
```

**Positional arguments:**

**inputFasta** Location of the input FASTA file

**outputDirectory** Location for the output files (Where *.bitmask and *.srprism files will be stored)

**Options:**

**--outputFilePrefix** Prefix for the output file name (default: inputFasta name, sans ".fasta" extension)

**--word_size=18** Database word size (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

### 1.3.3 assembly.py - *de novo* assembly

**This script contains a number of utilities for viral sequence assembly** from NGS reads. Primarily used for Lassa and Ebola virus analysis in the Sabeti Lab / Broad Institute Viral Genomics.

```
usage: assembly.py subcommand
```

**Sub-commands:**

**trim_rmdup_subsamp**

Take reads through Trimmomatic, Prinseq, and subsampling. This should probably move over to read_utils.

```
usage: assembly.py trim_rmdup_subsamp [-h] [--n_reads N_READS]
                                      [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                      [--version] [--tmp_dir TMP_DIR]
                                      [--tmp_dirKeep]
                                      inBam clipDb outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, unaligned BAM format. |
| **clipDb** | Trimmomatic clip DB. |
| **outBam** | Output reads, unaligned BAM format (currently, read groups and other header information are destroyed in this process). |

**Options:**

| | |
|---|---|
| **--n_reads=100000** | Subsample reads to no more than this many individual reads. Note that paired reads are given priority, and unpaired reads are included to reach the count if there are too few paired reads to reach n_reads. (default %(default)s) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**assemble_trinity**

This step runs the Trinity assembler. First trim reads with trimmomatic, rmdup with prinseq, and random subsample to no more than 100k reads.

```
usage: assembly.py assemble_trinity [-h] [--n_reads N_READS]
                                    [--outReads OUTREADS] [--always_succeed]
                                    [--JVMmemory JVMMEMORY]
                                    [--threads THREADS]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    inBam clipDb outFasta
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input unaligned reads, BAM format. |

**clipDb** Trimmomatic clip DB.

**outFasta** Output assembly.

**Options:**

**--n_reads=100000** Subsample reads to no more than this many pairs. (default %(default)s)

**--outReads** Save the trimmomatic/prinseq/subsamp reads to a BAM file

**--always_succeed=False** If Trinity fails (usually because insufficient reads to assemble), emit an empty fasta file as output. Default is to throw a DenovoAssemblyError.

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**assemble_spades**

De novo RNA-seq assembly with the SPAdes assembler.

```
usage: assembly.py assemble_spades [-h] [--contigsTrusted CONTIGS_TRUSTED]
                                   [--contigsUntrusted CONTIGS_UNTRUSTED]
                                   [--nReads N_READS] [--outReads OUTREADS]
                                   [--filterContigs] [--alwaysSucceed]
                                   [--minContigLen MIN_CONTIG_LEN]
                                   [--spadesOpts SPADES_OPTS]
                                   [--memLimitGb MEM_LIMIT_GB]
                                   [--threads THREADS]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                  in_bam clip_db out_fasta
```

**Positional arguments:**

**in_bam** Input unaligned reads, BAM format. May include both paired and unpaired reads.

**clip_db** Trimmomatic clip db

**out_fasta** Output assembled contigs. Note that, since this is RNA-seq assembly, for each assembled genomic region there may be several contigs representing different variants of that region.

**Options:**

**--contigsTrusted** Optional input contigs of high quality, previously assembled from the same sample

**--contigsUntrusted** Optional input contigs of high medium quality, previously assembled from the same sample

**--nReads=10000000** Before assembly, subsample the reads to at most this many

**--outReads** Save the trimmomatic/prinseq/subsamp reads to a BAM file

**--filterContigs=False** only output contigs SPAdes is sure of (drop lesser-quality contigs from output)

**--alwaysSucceed=False** if assembly fails for any reason, output an empty contigs file, rather than failing with an error code

**--minContigLen=0** only output contigs longer than this many bp

**--spadesOpts=** (advanced) Extra flags to pass to the SPAdes assembler

**--memLimitGb=4** Max memory to use, in GB (default: %(default)s)

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**gapfill_gap2seq**

This step runs the Gap2Seq tool to close gaps between contigs in a scaffold.

```
usage: assembly.py gapfill_gap2seq [-h] [--memLimitGb MEM_LIMIT_GB]
                                   [--timeSoftLimitMinutes TIME_SOFT_LIMIT_
↪MINUTES]
                                   [--maskErrors] [--gap2seqOpts GAP2SEQ_OPTS]
                                   [--randomSeed RANDOM_SEED]
                                   [--threads THREADS]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   in_scaffold in_bam out_scaffold
```

**Positional arguments:**

**in_scaffold** FASTA file containing the scaffold. Each FASTA record corresponds to one segment (for multi-segment genomes). Contigs within each segment are separated by Ns.

**in_bam** Input unaligned reads, BAM format.

**out_scaffold** Output assembly.

**Options:**

**--memLimitGb=4.0** Max memory to use, in gigabytes %(default)s

**--timeSoftLimitMinutes=60.0** Stop trying to close more gaps after this many minutes (default: %(default)s); this is a soft/advisory limit

**--maskErrors=False** In case of any error, just copy in_scaffold to out_scaffold, emulating a successful run that simply could not fill any gaps.

**--gap2seqOpts=** (advanced) Extra command-line options to pass to Gap2Seq

**--randomSeed=0** Random seed; 0 means use current time

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**order_and_orient**

This step cleans up the de novo assembly with a known reference genome. Uses MUMmer (nucmer or promer) to create a reference-based consensus sequence of aligned contigs (with runs of N's in between the de novo contigs).

```
usage: assembly.py order_and_orient [-h]
                                    [--outAlternateContigs␣
↪OUTALTERNATECONTIGS]
                                    [--nGenomeSegments N_GENOME_SEGMENTS]
                                    [--outReference OUTREFERENCE]
                                    [--outStats OUTSTATS]
                                    [--breaklen BREAKLEN] [--maxgap MAXGAP]
                                    [--minmatch MINMATCH]
                                    [--mincluster MINCLUSTER]
                                    [--min_pct_id MIN_PCT_ID]
                                    [--min_contig_len MIN_CONTIG_LEN]
                                    [--min_pct_contig_aligned MIN_PCT_CONTIG_
↪ALIGNED]
                                    [--threads THREADS]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    inFasta inReference [inReference ...]
                                    outFasta
```

**Positional arguments:**

**inFasta** Input de novo assembly/contigs, FASTA format.

**inReference** Reference genome for ordering, orienting, and merging contigs, FASTA format. Multiple filenames may be listed, each containing one reference genome. Alternatively, multiple references may be given by specifying a single filename, and giving the number of reference segments with the nGenomeSegments parameter. If multiple references are given, they must all contain the same number of segments listed in the same order.

**outFasta** Output assembly, FASTA format, with the same number of chromosomes as inReference, and in the same order.

**Options:**

**--outAlternateContigs** Output sequences (FASTA format) from alternative contigs that mapped, but were not chosen for the final output.

**--nGenomeSegments=0** Number of genome segments. If 0 (the default), the 'inReference' parameter is treated as one genome. If positive, the 'inReference' parameter is treated as a list of genomes of nGenomeSegments each.

**--outReference** Output the reference chosen for scaffolding to this file

**--outStats** Output stats used in reference selection

**--breaklen, -b** Amount to extend alignment clusters by (if –extend). nucmer default 200, promer default 60.

**--maxgap=200, -g=200** Maximum gap between two adjacent matches in a cluster. Our default is %(default)s. nucmer default 90, promer default 30. Manual suggests going to 1000.

**--minmatch=10, -l=10** Minimum length of an maximal exact match. Our default is %(default)s. nucmer default 20, promer default 6.

**--mincluster, -c** Minimum cluster length. nucmer default 65, promer default 20.

**--min_pct_id=0.6, -i=0.6** show-tiling: minimum percent identity for contig alignment (0.0 - 1.0, default: %(default)s)

**--min_contig_len=200** show-tiling: reject contigs smaller than this (default: %(default)s)

**--min_pct_contig_aligned=0.3, -v=0.3** show-tiling: minimum percent of contig length in alignment (0.0 - 1.0, default: %(default)s)

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**impute_from_reference**

This takes a de novo assembly, aligns against a reference genome, and imputes all missing positions (plus some of the chromosome ends) with the reference genome. This provides an assembly with the proper structure (but potentially wrong sequences in areas) from which we can perform further read-based refinement. Two steps: filter_short_seqs: We then toss out all assemblies that come out to < 15kb or < 95% unambiguous and fail otherwise. modify_contig: Finally, we trim off anything at the end that exceeds the length of the known reference assembly. We also replace all Ns and everything within 55bp of the chromosome ends with the reference sequence. This is clearly incorrect consensus sequence, but it allows downstream steps to map reads in parts of the genome that would otherwise be Ns, and we will correct all of the inferred positions with two steps of read-based refinement (below), and revert positions back to Ns where read support is lacking. FASTA indexing: output assembly is indexed for Picard, Samtools, Novoalign.

```
usage: assembly.py impute_from_reference [-h] [--newName NEWNAME]
                                         [--minLengthFraction␣
↪MINLENGTHFRACTION]
                                         [--minUnambig MINUNAMBIG]
                                         [--replaceLength REPLACELENGTH]
                                         [--aligner {muscle,mafft,mummer}]
                                         [--index]
                                         [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                         [--version] [--tmp_dir TMP_DIR]
                                         [--tmp_dirKeep]
                                        inFasta inReference outFasta
```

**Positional arguments:**

| | |
|---|---|
| **inFasta** | Input assembly/contigs, FASTA format, already ordered, oriented and merged with inReference. |
| **inReference** | Reference genome to impute with, FASTA format. |
| **outFasta** | Output assembly, FASTA format. |

**Options:**

| | |
|---|---|
| **--newName** | rename output chromosome (default: do not rename) |
| **--minLengthFraction=0.5** | minimum length for contig, as fraction of reference (default: %(default)s) |
| **--minUnambig=0.5** | minimum percentage unambiguous bases for contig (default: %(default)s) |
| **--replaceLength=0** | length of ends to be replaced with reference (default: %(default)s) |
| **--aligner=muscle** | which method to use to align inFasta to inReference. "muscle" = MUSCLE, "mafft" = MAFFT, "mummer" = nucmer. [default: %(default)s] |
| | Possible choices: muscle, mafft, mummer |
| **--index=False** | Index outFasta for Picard/GATK, Samtools, and Novoalign. |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s] |
| | Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**refine_assembly**

This a refinement step where we take a crude assembly, align all reads back to it, and modify the assembly to the majority allele at each position based on read pileups. This step considers both SNPs as well as indels called by GATK and will correct the consensus based on GATK calls. Reads are aligned with Novoalign, then PCR duplicates are removed with Picard (in order to debias the allele counts in the pileups), and realigned with GATK's IndelRealigner (in order to call indels). Output FASTA file is indexed for Picard, Samtools, and Novoalign.

```
usage: assembly.py refine_assembly [-h]
                                   [--already_realigned_bam ALREADY_REALIGNED_
↪BAM]
                                   [--outBam OUTBAM] [--outVcf OUTVCF]
                                   [--min_coverage MIN_COVERAGE]
                                   [--major_cutoff MAJOR_CUTOFF]
                                   [--novo_params NOVO_PARAMS]
                                   [--chr_names [CHR_NAMES [CHR_NAMES ...]]]
                                   [--keep_all_reads] [--JVMmemory JVMMEMORY]
                                   [--GATK_PATH GATK_PATH]
                                   [--NOVOALIGN_LICENSE_PATH NOVOALIGN_
↪LICENSE_PATH]
                                   [--threads THREADS]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   inFasta inBam outFasta
```

**Positional arguments:**

**inFasta** Input assembly, FASTA format, pre-indexed for Picard, Samtools, and Novoalign.

**inBam** Input reads, unaligned BAM format.

**outFasta** Output refined assembly, FASTA format, indexed for Picard, Samtools, and Novoalign.

**Options:**

**--already_realigned_bam** BAM with reads that are already aligned to inFasta. This bypasses the alignment process by novoalign and instead uses the given BAM to make an assembly. When set, outBam is ignored.

**--outBam** Reads aligned to inFasta. Unaligned and duplicate reads have been removed. GATK indel realigned.

**--outVcf** GATK genotype calls for genome in inFasta coordinate space.

**--min_coverage=3** Minimum read coverage required to call a position unambiguous.

**--major_cutoff=0.5** If the major allele is present at a frequency higher than this cutoff, we will call an unambiguous base at that position. If it is equal to or below this cutoff, we will call an ambiguous base representing all possible alleles at that position. [default: %(default)s]

**--novo_params=-r Random -l 40 -g 40 -x 20 -t 100** Alignment parameters for Novoalign.

**--chr_names=[]** Rename all output chromosomes (default: retain original chromosome names)

**--keep_all_reads=False** Retain all reads in BAM file? Default is to remove unaligned and duplicate reads.

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--GATK_PATH** A path containing the GATK jar file. This overrides the GATK_ENV environment variable or the GATK conda package. (default: %(default)s)

**--NOVOALIGN_LICENSE_PATH** A path to the novoalign.lic file. This overrides the NOVOALIGN_LICENSE_PATH environment variable. (default: %(default)s)

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**filter_short_seqs**

Check sequences in inFile, retaining only those that are at least minLength

```
usage: assembly.py filter_short_seqs [-h] [-f FORMAT] [-of OUTPUT_FORMAT]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version]
                                     inFile minLength minUnambig outFile
```

**Positional arguments:**

**inFile** input sequence file

**minLength** minimum length for contig

**minUnambig** minimum percentage unambiguous bases for contig

**outFile** output file

**Options:**

**-f=fasta, --format=fasta** Format for input sequence (default: %(default)s)

**-of=fasta, --output-format=fasta** Format for output sequence (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**modify_contig**

Modifies an input contig. Depending on the options selected, can replace N calls with reference calls, replace ambiguous calls with reference calls, trim to the length of the reference, replace contig ends with reference calls, and trim leading and trailing Ns. Author: rsealfon.

```
usage: assembly.py modify_contig [-h] [-n NAME] [-cn] [-t] [-r5] [-r3]
                                 [-l REPLACE_LENGTH] [-f FORMAT] [-r] [-rn]
                                 [-ca] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                                 [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
```

```
                                [--version]
                                input output ref
```

**Positional arguments:**

| | |
|---|---|
| **input** | input alignment of reference and contig (should contain exactly 2 sequences) |
| **output** | Destination file for modified contigs |
| **ref** | reference sequence name (exact match required) |

**Options:**

| | |
|---|---|
| **-n, --name** | fasta header output name (default: existing header) |
| **-cn=False, --call-reference-ns=False** | should the reference sequence be called if there is an N in the contig and a more specific base in the reference (default: %(default)s) |
| **-t=False, --trim-ends=False** | should ends of contig.fasta be trimmed to length of reference (default: %(default)s) |
| **-r5=False, --replace-5ends=False** | should the 5'-end of contig.fasta be replaced by reference (default: %(default)s) |
| **-r3=False, --replace-3ends=False** | should the 3'-end of contig.fasta be replaced by reference (default: %(default)s) |
| **-l=10, --replace-length=10** | length of ends to be replaced (if replace-ends is yes) (default: %(default)s) |
| **-f=fasta, --format=fasta** | Format for input alignment (default: %(default)s) |
| **-r=False, --replace-end-gaps=False** | Replace gaps at the beginning and end of the sequence with reference sequence (default: %(default)s) |
| **-rn=False, --remove-end-ns=False** | Remove leading and trailing N's in the contig (default: %(default)s) |
| **-ca=False, --call-reference-ambiguous=False** | should the reference sequence be called if the contig seq is ambiguous and the reference sequence is more informative & consistant with the ambiguous base (ie Y->C) (default: %(default)s) |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**vcf_to_fasta**

Take input genotypes (VCF) and construct a consensus sequence (fasta) by using majority-read-count alleles in the VCF. Genotypes in the VCF will be ignored–we will use the allele with majority read support (or an ambiguity base if there is no clear majority). Uncalled positions will be emitted as N's. Author: dpark.

```
usage: assembly.py vcf_to_fasta [-h] [--trim_ends] [--min_coverage MIN_DP]
                                [--major_cutoff MAJOR_CUTOFF]
                                [--min_dp_ratio MIN_DP_RATIO]
                                [--name [NAME [NAME ...]]]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version]
                                inVcf outFasta
```

**Positional arguments:**

| | |
|---|---|
| **inVcf** | Input VCF file |
| **outFasta** | Output FASTA file |

**Options:**

| | |
|---|---|
| **--trim_ends=False** | If specified, we will strip off continuous runs of N's from the beginning and end of the sequences before writing to output. Interior N's will not be changed. |
| **--min_coverage=3** | Specify minimum read coverage (with full agreement) to make a call. [default: %(default)s] |
| **--major_cutoff=0.5** | If the major allele is present at a frequency higher than this cutoff, we will call an unambiguous base at that position. If it is equal to or below this cutoff, we will call an ambiguous base representing all possible alleles at that position. [default: %(default)s] |
| **--min_dp_ratio=0.0** | The input VCF file often reports two read depth values (DP)–one for the position as a whole, and one for the sample in question. We can optionally reject calls in which the sample read count is below a specified fraction of the total read count. This filter will not apply to any sites unless both DP values are reported. [default: %(default)s] |
| **--name=[]** | output sequence names (default: reference names in VCF file) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**trim_fasta**

Take input sequences (fasta) and trim any continuous sections of N's from the ends of them. Write trimmed sequences to an output fasta file.

```
usage: assembly.py trim_fasta [-h]
                              [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                              [--version]
                              inFasta outFasta
```

**Positional arguments:**

| | |
|---|---|
| **inFasta** | Input fasta file |
| **outFasta** | Output (trimmed) fasta file |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**deambig_fasta**

Take input sequences (fasta) and replace any ambiguity bases with a random unambiguous base from among the possibilities described by the ambiguity code. Write output to fasta file.

```
usage: assembly.py deambig_fasta [-h]
                                 [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                 [--version]
                                 inFasta outFasta
```

**Positional arguments:**

| | |
|---|---|
| **inFasta** | Input fasta file |
| **outFasta** | Output fasta file |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

### 1.3.4 interhost.py - species and population-level genetic variation

**This script contains a number of utilities for SNP calling, multi-alignment,** phylogenetics, etc.

```
usage: interhost.py subcommand
```

**Sub-commands:**

**snpEff**

Annotate variants in VCF file with translation consequences using snpEff.

```
usage: interhost.py snpEff [-h] [--emailAddress EMAILADDRESS]
                           [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                           [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                           [--version]
                           inVcf genomes [genomes ...] outVcf
```

**Positional arguments:**

| | |
|---|---|
| **inVcf** | Input VCF file |
| **genomes** | genome name (snpEff db name, or NCBI accessions) |
| **outVcf** | Output VCF file |

**Options:**

**--emailAddress** Your email address. To access the Genbank CoreNucleotide database, NCBI requires you to specify your email address with each request. In case of excessive usage of the E-utilities, NCBI will attempt to contact a user at the email address provided before blocking access.

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**align_mafft**

Run the mafft alignment on the input FASTA file.

```
usage: interhost.py align_mafft [-h] [--localpair | --globalpair]
                                [--preservecase] [--reorder]
                                [--gapOpeningPenalty GAPOPENINGPENALTY]
                                [--ep EP] [--verbose] [--outputAsClustal]
                                [--maxiters MAXITERS] [--threads THREADS]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version] [--tmp_dir TMP_DIR]
                                [--tmp_dirKeep]
                                inFastas [inFastas ...] outFile
```

**Positional arguments:**

**inFastas** Input FASTA files.

**outFile** Output file containing alignment result (default format: FASTA)

**Options:**

**--localpair** All pairwise alignments are computed with the Smith-Waterman algorithm.

**--globalpair** All pairwise alignments are computed with the Needleman-Wunsch algorithm.

**--preservecase** Preserve base or aa case, as well as symbols.

**--reorder** Output is ordered aligned rather than in the order of the input (default: %(default)s).

**--gapOpeningPenalty=1.53** Gap opening penalty (default: %(default)s).

**--ep** Offset (works like gap extension penalty).

**--verbose=False** Full output (default: %(default)s).

**--outputAsClustal** Write output file in Clustal format rather than FASTA

**--maxiters=0** Maximum number of refinement iterations (default: %(default)s). Note: if "–localpair" or "–globalpair" is specified this defaults to 1000.

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**multichr_mafft**

Run the mafft alignment on a series of chromosomes provided in sample-partitioned FASTA files. Output as FASTA. (i.e. file1.fasta would contain chr1, chr2, chr3; file2.fasta would also contain chr1, chr2, chr3)

```
usage: interhost.py multichr_mafft [-h] [--localpair | --globalpair]
                                   [--preservecase] [--reorder]
                                   [--gapOpeningPenalty GAPOPENINGPENALTY]
                                   [--ep EP] [--verbose] [--outputAsClustal]
                                   [--maxiters MAXITERS]
                                   [--outFilePrefix OUTFILEPREFIX]
                                   [--sampleRelationFile SAMPLERELATIONFILE]
                                   [--sampleNameListFile SAMPLENAMELISTFILE]
                                   [--threads THREADS]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   inFastas [inFastas ...] outDirectory
```

**Positional arguments:**

**inFastas** Input FASTA files.

**outDirectory** Location for the output files (default is cwd: %(default)s)

**Options:**

**--localpair** All pairwise alignments are computed with the Smith-Waterman algorithm.

**--globalpair** All pairwise alignments are computed with the Needleman-Wunsch algorithm.

**--preservecase** Preserve base or aa case, as well as symbols.

**--reorder** Output is ordered aligned rather than in the order of the input (default: %(default)s).

**--gapOpeningPenalty=1.53** Gap opening penalty (default: %(default)s).

**--ep** Offset (works like gap extension penalty).

**--verbose=False** Full output (default: %(default)s).

**--outputAsClustal** Write output file in Clustal format rather than FASTA

**--maxiters=0** Maximum number of refinement iterations (default: %(default)s). Note: if "–localpair" or "–globalpair" is specified this defaults to 1000.

**--outFilePrefix=aligned** Prefix for the output file name (default: %(default)s)

**--sampleRelationFile** If the parameter sampleRelationFile is specified (as a file path), a JSON file will be written mapping sample name to sequence position in the output.

**--sampleNameListFile** If the parameter sampleRelationFile is specified (as a file path), a file will be written mapping sample names in the order of their sequence positions in the output.

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

### 1.3.5 intrahost.py - within-host genetic variation (iSNVs)

This script contains a number of utilities for intrahost variant calling and annotation for viral genomes.

```
usage: intrahost.py subcommand
```

**Sub-commands:**

**vphaser_one_sample**

Input: a single BAM file, representing reads from one sample, mapped to its own consensus assembly. It may contain multiple read groups and libraries. Output: a tab-separated file with no header containing filtered V Phaser-2 output variants with additional column for sequence/chrom name, and library counts and p-values appended to the counts for each allele.

```
usage: intrahost.py vphaser_one_sample [-h]
                                       [--vphaserNumThreads VPHASERNUMTHREADS]
                                       [--minReadsEach MINREADSEACH]
                                       [--maxBias MAXBIAS]
                                       [--removeDoublyMappedReads]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version]
                                       inBam inConsFasta outTab
```

**Positional arguments:**

**inBam** Input Bam file.

**inConsFasta** Consensus assembly fasta.

**outTab** Tab-separated headerless output file.

**Options:**

**--vphaserNumThreads** Number of threads in call to V-Phaser 2.

**--minReadsEach=5** Minimum number of reads on each strand (default: %(default)s).

**--maxBias=10** Maximum allowable ratio of number of reads on the two strands (default: %(default)s). Ignored if minReadsEach = 0.

**--removeDoublyMappedReads=False** When calling V-Phaser, remove reads mapping to more than one contig. Default is to keep the reads.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**vphaser**

Run V-Phaser 2 on the input file without any additional filtering. Combine the non-header lines of the CHROM.var.raw.txt files it produces, adding CHROM as the first field on each line.

```
usage: intrahost.py vphaser [-h] [--numThreads NUMTHREADS]
                            [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                            [--version]
                            inBam outTab
```

**Positional arguments:**

**inBam** Input Bam file.

**outTab** Tab-separated headerless output file.

**Options:**

**--numThreads** Number of threads in call to V-Phaser 2.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**tabfile_rename**

Take input tab file and copy to an output file while changing the values in a specific column based on a mapping file. The first line will pass through untouched (it is assumed to be a header).

```
usage: intrahost.py tabfile_rename [-h] [--col_idx COL]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version]
                                   inFile mapFile outFile
```

**Positional arguments:**

**inFile** Input flat file

**mapFile** Map file. Two-column headerless file that maps input values to output values. This script will error if there are values in inFile that do not exist in mapFile.

**outFile** Output flat file

**Options:**

**--col_idx=0** Which column number to replace (0-based index). [default: %(default)s]

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**merge_to_vcf**

Combine and convert vPhaser2 parsed filtered output text files into VCF format. Assumption: consensus assemblies used in creating alignments do not extend beyond ends of reference. the number of alignment files equals the number of chromosomes / segments

```
usage: intrahost.py merge_to_vcf [-h] [--samples [SAMPLES [SAMPLES ...]]]
                                 --isnvs ISNVS [ISNVS ...] --alignments
                                 ALIGNMENTS [ALIGNMENTS ...]
                                 [--strip_chr_version] [--naive_filter]
                                 [--parse_accession]
                                 [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                 [--version]
                                 refFasta outVcf
```

**Positional arguments:**

**refFasta** The target reference genome. outVcf will use these chromosome names, coordinate spaces, and reference alleles

**outVcf** Output VCF file containing all variants

**Options:**

**--samples** A list of sample names

**--isnvs** A list of file names from the output of vphaser_one_sample These must be in the SAME ORDER as samples.

**--alignments** a list of fasta files containing multialignment of input assemblies, with one file per chromosome/segment. Each alignment file will contain a line for each sample, as well as the reference genome to which they were aligned.

**--strip_chr_version=False** If set, strip any trailing version numbers from the chromosome names. If the chromosome name ends with a period followed by integers, this is interepreted as a version number to be removed. This is because Genbank accession numbers are often used by SnpEff databases downstream, but without the corresponding version number. Default is false (leave chromosome names untouched).

**--naive_filter=False** If set, keep only the alleles that have at least two independent libraries of support and allele freq > 0.005. Default is false (do not filter at this stage).

**--parse_accession=False** If set, parse only the accession for the chromosome name. Helpful if snpEff has to create its own database

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**Fws**

Compute the Fws statistic on iSNV data. See Manske, 2012 (Nature)

```
usage: intrahost.py Fws [-h]
                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                        [--version]
                        inVcf outVcf
```

**Positional arguments:**

**inVcf** Input VCF file

**outVcf** Output VCF file

**Options:**

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**iSNV_table**

Convert VCF iSNV data to tabular text

```
usage: intrahost.py iSNV_table [-h]
                               [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                               [--version]
                               inVcf outFile
```

**Positional arguments:**

**inVcf** Input VCF file

**outFile** Output text file

**Options:**

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**iSNP_per_patient**

Aggregate tabular iSNP data per patient x position (all time points averaged)

```
usage: intrahost.py iSNP_per_patient [-h]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version]
                                     inFile outFile
```

**Positional arguments:**

| | |
|---|---|
| **inFile** | Input text file |
| **outFile** | Output text file |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

### 1.3.6 read_utils.py - utilities that manipulate bam and fastq files

Utilities for working with sequence reads, such as converting formats and fixing mate pairs.

```
usage: read_utils.py subcommand
```

**Sub-commands:**

**purge_unmated**

Use mergeShuffledFastqSeqs to purge unmated reads, and put corresponding reads in the same order. Corresponding sequences must have sequence identifiers of the form SEQID/1 and SEQID/2.

```
usage: read_utils.py purge_unmated [-h] [--regex REGEX]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   inFastq1 inFastq2 outFastq1 outFastq2
```

**Positional arguments:**

| | |
|---|---|
| **inFastq1** | Input fastq file; 1st end of paired-end reads. |
| **inFastq2** | Input fastq file; 2nd end of paired-end reads. |
| **outFastq1** | Output fastq file; 1st end of paired-end reads. |
| **outFastq2** | Output fastq file; 2nd end of paired-end reads. |

**Options:**

| | |
|---|---|
| **--regex=^@(\S+)/[1\|2]$** | Perl regular expression to parse paired read IDs (default: %(default)s) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**index_fasta_samtools**

Index a reference genome for Samtools.

```
usage: read_utils.py index_fasta_samtools [-h]
                                          [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                          [--version]
                                          inFasta
```

**Positional arguments:**

| | |
|---|---|
| **inFasta** | Reference genome, FASTA format. |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**index_fasta_picard**

Create an index file for a reference genome suitable for Picard/GATK.

```
usage: read_utils.py index_fasta_picard [-h] [--JVMmemory JVMMEMORY]
                                        [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                        [--version] [--tmp_dir TMP_DIR]
                                        [--tmp_dirKeep]
                                        inFasta
```

**Positional arguments:**

| | |
|---|---|
| **inFasta** | Input reference genome, FASTA format. |

**Options:**

| | |
|---|---|
| **--JVMmemory=512m** | JVM virtual memory size (default: %(default)s) |
| **--picardOptions=[]** | Optional arguments to Picard's CreateSequenceDictionary, OPTIONNAME=value ... |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**mkdup_picard**

Mark or remove duplicate reads from BAM file.

```
usage: read_utils.py mkdup_picard [-h] [--outMetrics OUTMETRICS] [--remove]
                                  [--JVMmemory JVMMEMORY]
                                  [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep]
                                  inBams [inBams ...] outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBams** | Input reads, BAM format. |
| **outBam** | Output reads, BAM format. |

**Options:**

| | |
|---|---|
| **--outMetrics** | Output metrics file. Default is to dump to a temp file. |
| **--remove=False** | Instead of marking duplicates, remove them entirely (default: %(default)s) |
| **--JVMmemory=2g** | JVM virtual memory size (default: %(default)s) |
| **--picardOptions=[]** | Optional arguments to Picard's MarkDuplicates, OPTION-NAME=value ... |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**revert_bam_picard**

Revert BAM to raw reads

```
usage: read_utils.py revert_bam_picard [-h] [--JVMmemory JVMMEMORY]
                                       [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                       [--clearTags]
                                       [--tagsToClear TAGS_TO_CLEAR [TAGS_TO_
↪CLEAR ...]]
                                       [--doNotSanitize]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version] [--tmp_dir TMP_DIR]
                                       [--tmp_dirKeep]
                                       inBam outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, BAM format. |
| **outBam** | Output reads, BAM format. |

**Options:**

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--picardOptions=[]** Optional arguments to Picard's RevertSam, OPTIONNAME=value ...

**--clearTags=False** When supplying an aligned input file, clear the per-read attribute tags

**--tagsToClear=['XT', 'X0', 'X1', 'XA', 'AM', 'SM', 'BQ', 'CT', 'XN', 'OC', 'OP']** A space-separated list of tags to remove from all reads in the input bam file (default: %(default)s)

**--doNotSanitize=False** Undocumented

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**picard**

Generic Picard runner.

```
usage: read_utils.py picard [-h] [--JVMmemory JVMMEMORY]
                            [--picardOptions [PICARDOPTIONS [PICARDOPTIONS ...
↪]]]
                            [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                            [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                            command
```

**Positional arguments:**

**command** picard command

**Options:**

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--picardOptions=[]** Optional arguments to Picard, OPTIONNAME=value ...

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**sort_bam**

Sort BAM file

```
usage: read_utils.py sort_bam [-h] [--index] [--md5] [--JVMmemory JVMMEMORY]
                              [--picardOptions [PICARDOPTIONS [PICARDOPTIONS .
↪..]]]
                              [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                              [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                              inBam outBam {unsorted,queryname,coordinate}
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input bam file. |
| **outBam** | Output bam file, sorted. |
| **sortOrder** | How to sort the reads. [default: %(default)s]<br>Possible choices: unsorted, queryname, coordinate |

**Options:**

| | |
|---|---|
| **--index=False** | Index outBam (default: %(default)s) |
| **--md5=False** | MD5 checksum outBam (default: %(default)s) |
| **--JVMmemory=2g** | JVM virtual memory size (default: %(default)s) |
| **--picardOptions=[]** | Optional arguments to Picard's SortSam, OPTIONNAME=value ... |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**downsample_bams**

Downsample multiple bam files to the smallest read count in common, or to the specified count.

```
usage: read_utils.py downsample_bams [-h] [--outPath OUT_PATH]
                                     [--readCount SPECIFIED_READ_COUNT]
                                     [--deduplicateBefore | --
↪deduplicateAfter]
                                     [--JVMmemory JVMMEMORY]
                                     [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                     [--threads THREADS]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version] [--tmp_dir TMP_DIR]
                                     [--tmp_dirKeep]
                                     in_bams [in_bams ...]
```

**Positional arguments:**

| | |
|---|---|
| **in_bams** | Input bam files. |

**Options:**

**--outPath** Output path. If not provided, downsampled bam files will be written to the same paths as each source bam file

**--readCount** The number of reads to downsample to.

**--deduplicateBefore=False** de-duplicate reads before downsampling.

**--deduplicateAfter=False** de-duplicate reads after downsampling.

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--picardOptions=[]** Optional arguments to Picard's DownsampleSam, OPTION-NAME=value ...

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**merge_bams**

Merge multiple BAMs into one

```
usage: read_utils.py merge_bams [-h] [--JVMmemory JVMMEMORY]
                                [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version] [--tmp_dir TMP_DIR]
                                [--tmp_dirKeep]
                                inBams [inBams ...] outBam
```

**Positional arguments:**

**inBams** Input bam files.

**outBam** Output bam file.

**Options:**

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--picardOptions=[]** Optional arguments to Picard's MergeSamFiles, OPTION-NAME=value ...

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

### filter_bam

Filter BAM file by read name

```
usage: read_utils.py filter_bam [-h] [--exclude] [--JVMmemory JVMMEMORY]
                                [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version] [--tmp_dir TMP_DIR]
                                [--tmp_dirKeep]
                                inBam readList outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input bam file. |
| **readList** | Input file of read IDs. |
| **outBam** | Output bam file. |

**Options:**

| | |
|---|---|
| **--exclude=False** | If specified, readList is a list of reads to remove from input. Default behavior is to treat readList as an inclusion list (all unnamed reads are removed). |
| **--JVMmemory=4g** | JVM virtual memory size (default: %(default)s) |
| **--picardOptions=[]** | Optional arguments to Picard's FilterSamReads, OPTION-NAME=value ... |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

### fastq_to_bam

Convert a pair of fastq paired-end read files and optional text header to a single bam file.

```
usage: read_utils.py fastq_to_bam [-h]
                                  (--sampleName SAMPLENAME | --header HEADER)
                                  [--JVMmemory JVMMEMORY]
                                  [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep]
                                  inFastq1 inFastq2 outBam
```

**Positional arguments:**

| | |
|---|---|
| **inFastq1** | Input fastq file; 1st end of paired-end reads. |
| **inFastq2** | Input fastq file; 2nd end of paired-end reads. |

**outBam** Output bam file.

**Options:**

**--sampleName** Sample name to insert into the read group header.

**--header** Optional text file containing header.

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--picardOptions=[]** Optional arguments to Picard's FastqToSam, OPTION-NAME=value ... Note that header-related options will be overwritten by HEADER if present.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**join_paired_fastq**

Join paired fastq reads into single reads with Ns

```
usage: read_utils.py join_paired_fastq [-h] [--outFormat OUTFORMAT]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version] [--tmp_dir TMP_DIR]
                                       [--tmp_dirKeep]
                                       output inFastqs [inFastqs ...]
```

**Positional arguments:**

**output** Output file.

**inFastqs** Input fastq file (2 if paired, 1 if interleaved)

**Options:**

**--outFormat=fastq** Output file format.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**split_bam**

Split BAM file equally into several output BAM files.

```
usage: read_utils.py split_bam [-h]
                               [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
```

```
                                [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                                inBam outBams [outBams ...]
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input BAM file. |
| **outBams** | Output BAM files |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**reheader_bam**

Copy a BAM file (inBam to outBam) while renaming elements of the BAM header. The mapping file specifies which (key, old value, new value) mappings. For example: LB lib1 lib_one SM sample1 Sample_1 SM sample2 Sample_2 SM sample3 Sample_3 CN broad BI

```
usage: read_utils.py reheader_bam [-h]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep]
                                  inBam rgMap outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, BAM format. |
| **rgMap** | Tabular file containing three columns: field, old, new. |
| **outBam** | Output reads, BAM format. |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**reheader_bams**

Copy BAM files while renaming elements of the BAM header. The mapping file specifies which (key, old value, new value) mappings. For example: LB lib1 lib_one SM sample1 Sample_1 SM sample2 Sample_2 SM sample3 Sample_3 CN broad BI FN in1.bam out1.bam FN in2.bam out2.bam

```
usage: read_utils.py reheader_bams [-h]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   rgMap
```

**Positional arguments:**

| | |
|---|---|
| **rgMap** | Tabular file containing three columns: field, old, new. |

**Options:**

| | |
|---|---|
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**rmdup_cdhit_bam**

Remove duplicate reads from BAM file using cd-hit-dup.

```
usage: read_utils.py rmdup_cdhit_bam [-h] [--JVMmemory JVM_MEMORY]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version] [--tmp_dir TMP_DIR]
                                     [--tmp_dirKeep]
                                     inBam outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, BAM format. |
| **outBam** | Output reads, BAM format. |

**Options:**

| | |
|---|---|
| **--JVMmemory=4g** | JVM virtual memory size (default: %(default)s) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**rmdup_mvicuna_bam**

Remove duplicate reads from BAM file using M-Vicuna. The primary advantage to this approach over Picard's MarkDuplicates tool is that Picard requires that input reads are aligned to a reference, and M-Vicuna can operate on unaligned reads.

```
usage: read_utils.py rmdup_mvicuna_bam [-h] [--JVMmemory JVMMEMORY]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version] [--tmp_dir TMP_DIR]
                                       [--tmp_dirKeep]
                                       inBam outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, BAM format. |
| **outBam** | Output reads, BAM format. |

**Options:**

| | |
|---|---|
| **--JVMmemory=4g** | JVM virtual memory size (default: %(default)s) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**rmdup_prinseq_fastq**

Run prinseq-lite's duplicate removal operation on paired-end reads. Also removes reads with more than one N.

```
usage: read_utils.py rmdup_prinseq_fastq [-h] [--includeUnmated]
                                         [--unpairedOutFastq1␣
↪UNPAIREDOUTFASTQ1]
                                         [--unpairedOutFastq2␣
↪UNPAIREDOUTFASTQ2]
                                         [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                         [--version] [--tmp_dir TMP_DIR]
                                         [--tmp_dirKeep]
                                         inFastq1 inFastq2 outFastq1 outFastq2
```

**Positional arguments:**

| | |
|---|---|
| **inFastq1** | Input fastq file; 1st end of paired-end reads. |
| **inFastq2** | Input fastq file; 2nd end of paired-end reads. |
| **outFastq1** | Output fastq file; 1st end of paired-end reads. |
| **outFastq2** | Output fastq file; 2nd end of paired-end reads. |

**Options:**

| | |
|---|---|
| **--includeUnmated=False** | Include unmated reads in the main output fastq files (default: %(default)s) |
| **--unpairedOutFastq1** | File name of output unpaired reads from 1st end of paired-end reads (independent of –includeUnmated) |

**--unpairedOutFastq2** File name of output unpaired reads from 2nd end of paired-end reads (independent of –includeUnmated)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**filter_bam_mapped_only**

Samtools to reduce a BAM file to only reads that are aligned (-F 4) with a non-zero mapping quality (-q 1) and are not marked as a PCR/optical duplicate (-F 1024).

```
usage: read_utils.py filter_bam_mapped_only [-h]
                                            [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                            [--version] [--tmp_dir TMP_DIR]
                                            [--tmp_dirKeep]
                                            inBam outBam
```

**Positional arguments:**

**inBam** Input aligned reads, BAM format.

**outBam** Output sorted indexed reads, filtered to aligned-only, BAM format.

**Options:**

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**novoalign**

Align reads with Novoalign. Sort and index BAM output.

```
usage: read_utils.py novoalign [-h] [--options OPTIONS] [--min_qual MIN_QUAL]
                               [--JVMmemory JVMMEMORY]
                               [--NOVOALIGN_LICENSE_PATH NOVOALIGN_LICENSE_
↪PATH]
                               [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                               [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                               inBam refFasta outBam
```

**Positional arguments:**

**inBam** Input reads, BAM format.

**refFasta** Reference genome, FASTA format, pre-indexed by Novoindex.

**outBam** Output reads, BAM format (aligned).

**Options:**

**--options=-r Random** Novoalign options (default: %(default)s)

**--min_qual=0** Filter outBam to minimum mapping quality (default: %(default)s)

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--NOVOALIGN_LICENSE_PATH** A path to the novoalign.lic file. This overrides the NOVOALIGN_LICENSE_PATH environment variable. (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**novoindex**

```
usage: read_utils.py novoindex [-h]
                               [--NOVOALIGN_LICENSE_PATH NOVOALIGN_LICENSE_
↪PATH]
                               [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                               [--version]
                               refFasta
```

**Positional arguments:**

**refFasta** Reference genome, FASTA format.

**Options:**

**--NOVOALIGN_LICENSE_PATH** A path to the novoalign.lic file. This overrides the NOVOALIGN_LICENSE_PATH environment variable. (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**gatk_ug**

Call genotypes using the GATK UnifiedGenotyper.

```
usage: read_utils.py gatk_ug [-h] [--options OPTIONS] [--JVMmemory JVMMEMORY]
                             [--GATK_PATH GATK_PATH]
                             [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
```

```
                                [--version] [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                                inBam refFasta outVcf
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, BAM format. |
| **refFasta** | Reference genome, FASTA format, pre-indexed by Picard. |
| **outVcf** | Output calls in VCF format. If this filename ends with .gz, GATK will BGZIP compress the output and produce a Tabix index file as well. |

**Options:**

| | |
|---|---|
| **--options=–min_base_quality_score 15 -ploidy 4** | UnifiedGenotyper options (default: %(default)s) |
| **--JVMmemory=2g** | JVM virtual memory size (default: %(default)s) |
| **--GATK_PATH** | A path containing the GATK jar file. This overrides the GATK_ENV environment variable or the GATK conda package. (default: %(default)s) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

**gatk_realign**

Local realignment of BAM files with GATK IndelRealigner.

```
usage: read_utils.py gatk_realign [-h] [--JVMmemory JVMMEMORY]
                                  [--GATK_PATH GATK_PATH]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep] [--threads THREADS]
                                  inBam refFasta outBam
```

**Positional arguments:**

| | |
|---|---|
| **inBam** | Input reads, BAM format, aligned to refFasta. |
| **refFasta** | Reference genome, FASTA format, pre-indexed by Picard. |
| **outBam** | Realigned reads. |

**Options:**

| | |
|---|---|
| **--JVMmemory=2g** | JVM virtual memory size (default: %(default)s) |
| **--GATK_PATH** | A path containing the GATK jar file. This overrides the GATK_ENV environment variable or the GATK conda package. (default: %(default)s) |

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**--threads** Number of threads (default: all available cores)

**align_and_fix**

Take reads, align to reference with Novoalign, optionally mark duplicates with Picard, realign indels with GATK, and optionally filters final file to mapped/non-dupe reads.

```
usage: read_utils.py align_and_fix [-h] [--outBamAll OUTBAMALL]
                                   [--outBamFiltered OUTBAMFILTERED]
                                   [--aligner_options ALIGNER_OPTIONS]
                                   [--aligner {novoalign,bwa}]
                                   [--JVMmemory JVMMEMORY] [--threads THREADS]
                                   [--skipMarkDupes] [--GATK_PATH GATK_PATH]
                                   [--NOVOALIGN_LICENSE_PATH NOVOALIGN_
↪LICENSE_PATH]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   inBam refFasta
```

**Positional arguments:**

**inBam** Input unaligned reads, BAM format.

**refFasta** Reference genome, FASTA format; will be indexed by Picard and Novoalign.

**Options:**

**--outBamAll** Aligned, sorted, and indexed reads. Unmapped and duplicate reads are retained. By default, duplicate reads are marked. If "–skipMarkDupes" is specified duplicate reads are included in outout without being marked.

**--outBamFiltered** Aligned, sorted, and indexed reads. Unmapped reads are removed from this file, as well as any marked duplicate reads. Note that if "–skipMarkDupes" is provided, duplicates will be not be marked and will be included in the output.

**--aligner_options** aligner options (default for novoalign: "-r Random", bwa: "-T 30"

**--aligner=novoalign** aligner (default: %(default)s)

Possible choices: novoalign, bwa

**--JVMmemory=4g** JVM virtual memory size (default: %(default)s)

**--threads** Number of threads (default: all available cores)

**--skipMarkDupes=False** If specified, duplicate reads will not be marked in the resulting output file.

**--GATK_PATH** A path containing the GATK jar file. This overrides the GATK_ENV environment variable or the GATK conda package. (default: %(default)s)

**--NOVOALIGN_LICENSE_PATH** A path to the novoalign.lic file. This overrides the NOVOALIGN_LICENSE_PATH environment variable. (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**bwamem_idxstats**

Take reads, align to reference with BWA-MEM and perform samtools idxstats.

```
usage: read_utils.py bwamem_idxstats [-h] [--outBam OUTBAM]
                                     [--outStats OUTSTATS]
                                     [--minScoreToFilter MIN_SCORE_TO_FILTER]
                                     [--alignerOptions ALIGNER_OPTIONS]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version] [--tmp_dir TMP_DIR]
                                     [--tmp_dirKeep]
                                     inBam refFasta
```

**Positional arguments:**

**inBam** Input unaligned reads, BAM format.

**refFasta** Reference genome, FASTA format, pre-indexed by Picard and Novoalign.

**Options:**

**--outBam** Output aligned, indexed BAM file

**--outStats** Output idxstats file

**--minScoreToFilter** Filter bwa alignments using this value as the minimum allowed alignment score. Specifically, sum the alignment scores across all alignments for each query (including reads in a pair, supplementary and secondary alignments) and then only include, in the output, queries whose summed alignment score is at least this value. This is only applied when the aligner is 'bwa'. The filtering on a summed alignment score is sensible for reads in a pair and supplementary alignments, but may not be reasonable if bwa outputs secondary alignments (i.e., if '-a' is in the aligner options). (default: not set - i.e., do not filter bwa's output)

**--alignerOptions** bwa options (default: bwa defaults)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**extract_tarball**

Extract an input .tar, .tgz, .tar.gz, .tar.bz2, .tar.lz4, or .zip file to a given directory (or we will choose one on our own). Emit the resulting directory path to stdout.

```
usage: read_utils.py extract_tarball [-h]
                                     [--compression {gz,bz2,lz4,zip,none,auto}
↪]
                                     [--pipe_hint PIPE_HINT]
                                     [--threads THREADS]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version] [--tmp_dir TMP_DIR]
                                     [--tmp_dirKeep]
                                    tarfile out_dir
```

**Positional arguments:**

**tarfile** Input tar file. May be "-" for stdin.

**out_dir** Output directory

**Options:**

**--compression=auto** Compression type (default: %(default)s). Auto-detect is incompatible with stdin input unless pipe_hint is specified.

Possible choices: gz, bz2, lz4, zip, none, auto

**--pipe_hint** If tarfile is stdin, you can provide a file-like URI string for pipe_hint which ends with a common compression file extension if you want to use compression=auto.

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**read_names**

Extract read names from a sequence file

```
usage: read_utils.py read_names [-h] [--threads THREADS]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version] [--tmp_dir TMP_DIR]
                                [--tmp_dirKeep]
                                in_reads out_read_names
```

**Positional arguments:**

| | |
|---|---|
| **in_reads** | the input reads ([compressed] fasta or bam) |
| **out_read_names** | the read names |

**Options:**

| | |
|---|---|
| **--threads** | Number of threads (default: all available cores) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

### 1.3.7 kmer_utils.py - utilities that manipulate kmer sets and use them to filter reads or reference sequences

Commands for working with sets of kmers

```
usage: kmer_utils.py subcommand
```

**Sub-commands:**

**build_kmer_db**

Build a database of kmers occurring in given sequences.

```
usage: kmer_utils.py build_kmer_db [-h] [--kmerSize KMER_SIZE]
                                   [--minOccs MIN_OCCS] [--maxOccs MAX_OCCS]
                                   [--counterCap COUNTER_CAP] [--singleStrand]
                                   [--memLimitGb MEM_LIMIT_GB]
                                   [--memLimitLaxness {0,1,2}]
                                   [--threads THREADS]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   [--version] [--tmp_dir TMP_DIR]
                                   [--tmp_dirKeep]
                                   seq_files [seq_files ...] kmer_db
```

**Positional arguments:**

| | |
|---|---|
| **seq_files** | Files from which to extract kmers (fasta/fastq/bam, fasta/fastq may be .gz or .bz2) |
| **kmer_db** | kmer database (with or without .kmc_pre/.kmc_suf suffix) |

**Options:**

**--kmerSize=25, -k=25** kmer size

**--minOccs=1, -ci=1** drop kmers with fewer than this many occurrences

**--maxOccs=2147483647, -cx=2147483647** drop kmers with more than this many occurrences

**--counterCap=255, -cs=255** cap kmer counts at this value

**--singleStrand=False, -b=False** do not add kmers from reverse complements of input sequences

**--memLimitGb=8** Max memory to use, in GB

**--memLimitLaxness=0** How strict is –memLimitGb? 0=strict, 1=lax, 2=even more lax

Possible choices: 0, 1, 2

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**dump_kmer_counts**

Dump kmers and their counts from kmer database to a text file

```
usage: kmer_utils.py dump_kmer_counts [-h] [--minOccs MIN_OCCS]
                                      [--maxOccs MAX_OCCS] [--threads THREADS]
                                      [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                      [--version] [--tmp_dir TMP_DIR]
                                      [--tmp_dirKeep]
                                     kmer_db out_kmers
```

**Positional arguments:**

**kmer_db** kmer database (with or without .kmc_pre/.kmc_suf suffix)

**out_kmers** text file to which to write the kmers

**Options:**

**--minOccs=1, -ci=1** drop kmers with fewer than this many occurrences

**--maxOccs=2147483647, -cx=2147483647** drop kmers with more than this many occurrences

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**filter_reads**

Filter reads based on their kmer contents.

Can also be used to filter contigs or reference sequences, but we'll refer to filtering of reads in the documentation.

Note that "occurrence of a kmer" means "occurrence of the kmer or its reverse complement" if kmer_db was built with single_strand==False.

Inputs: kmer_db: the kmc kmer database in_reads: the reads to filter. can be a .fasta or .fastq or .bam; fasta or fastq can be compressed with gzip or bzip2. If a .bam, a read pair is kept if either mate passes the filter.

Outputs: out_reads: file to which filtered reads are written. type is determined from extension, same types as above are supported.

Params: db_min_occs: only consider database kmers with at least this count db_max_occs: only consider database kmers with at most this count

read_min_occs: only keep reads with at least this many occurrences of kmers from database. read_max_occs: only keep reads with no more than this many occurrence of kmers from the database. read_min_occs_frac: only keep reads with at least this many occurrences of kmers from database, interpreted as a fraction of read length in kmers read_max_occs_frac: only keep reads with no more than this many occurrence of kmers from the database. interpreted as a fraction of read length in kmers.

(Note: minimal read kmer content can be given as absolute counts or fraction of read length, but not both).

hard_mask: if True, in the output reads, kmers not passing the filter are replaced by Ns threads: use this many threads

```
usage: kmer_utils.py filter_reads [-h] [--dbMinOccs DB_MIN_OCCS]
                                  [--dbMaxOccs DB_MAX_OCCS]
                                  [--readMinOccs READ_MIN_OCCS]
                                  [--readMaxOccs READ_MAX_OCCS]
                                  [--readMinOccsFrac READ_MIN_OCCS_FRAC]
                                  [--readMaxOccsFrac READ_MAX_OCCS_FRAC]
                                  [--hardMask] [--threads THREADS]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep]
                                  kmer_db in_reads out_reads
```

**Positional arguments:**

**kmer_db** kmer database (with or without .kmc_pre/.kmc_suf suffix)

**in_reads** input reads, as fasta/fastq/bam

**out_reads** output reads

**Options:**

**--dbMinOccs=1** ignore datatbase kmers with count below this

**--dbMaxOccs=2147483647** ignore datatbase kmers with count above this

**--readMinOccs=0** filter out reads with fewer than this many db kmers

**--readMaxOccs=2147483647** filter out reads with more than this many db kmers

**--readMinOccsFrac=0.0** filter out reads with fewer than this many db kmers, interpreted as fraction of read length

**--readMaxOccsFrac=1.0** filter out reads with more than this many db kmers, interpreted as fraction of read length

**--hardMask=False** In the output reads, mask the invalid kmers

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**kmers_binary_op**

Perform a simple binary operation on kmer sets.

```
usage: kmer_utils.py kmers_binary_op [-h] [--resultMinOccs RESULT_MIN_OCCS]
                                     [--resultMaxOccs RESULT_MAX_OCCS]
                                     [--resultCounterCap RESULT_COUNTER_CAP]
                                     [--threads THREADS]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version] [--tmp_dir TMP_DIR]
                                     [--tmp_dirKeep]
                                     {intersect,union,kmers_subtract,counters_
↪subtract}
                                    kmer_db1 kmer_db2 kmer_db_out
```

**Positional arguments:**

**op** binary operation to perform

Possible choices: intersect, union, kmers_subtract, counters_subtract

**kmer_db1** first kmer set

**kmer_db2** second kmer set

**kmer_db_out** output kmer db

**Options:**

**--resultMinOccs=1** from the result drop kmers with counts below this

**--resultMaxOccs=2147483647** from the result drop kmers with counts above this

**--resultCounterCap=255** cap output counters at this value

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**kmers_set_counts**

Copy the kmer database, setting all kmer counts in the output to the given value.

```
usage: kmer_utils.py kmers_set_counts [-h] [--threads THREADS]
                                      [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                      [--version] [--tmp_dir TMP_DIR]
                                      [--tmp_dirKeep]
                                      kmer_db_in value kmer_db_out
```

**Positional arguments:**

**kmer_db_in** input kmer db

**value** all kmer counts in the output will be set to this value

**kmer_db_out** output kmer db

**Options:**

**--threads** Number of threads (default: all available cores)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

### 1.3.8 reports.py - produce various metrics and reports

Functions to create reports from genomics pipeline data.

```
usage: reports.py subcommand
```

**Sub-commands:**

**assembly_stats**

Fetch assembly-level statistics for a given sample

```
usage: reports.py assembly_stats [-h]
                                 [--cov_thresholds COV_THRESHOLDS [COV_
↪THRESHOLDS ...]]
                                 [--assembly_dir ASSEMBLY_DIR]
```

```
                                [--assembly_tmp ASSEMBLY_TMP]
                                [--align_dir ALIGN_DIR]
                                [--reads_dir READS_DIR]
                                [--raw_reads_dir RAW_READS_DIR]
                                samples [samples ...] outFile
```

**Positional arguments:**

**samples** Sample names.

**outFile** Output report file.

**Options:**

**--cov_thresholds=(1, 5, 20, 100)** Genome coverage thresholds to report on. (default: %(default)s)

**--assembly_dir=data/02_assembly** Directory with assembly outputs. (default: %(default)s)

**--assembly_tmp=tmp/02_assembly** Directory with assembly temp files. (default: %(default)s)

**--align_dir=data/02_align_to_self** Directory with reads aligned to own assembly. (default: %(default)s)

**--reads_dir=data/01_per_sample** Directory with unaligned filtered read BAMs. (default: %(default)s)

**--raw_reads_dir=data/00_raw** Directory with unaligned raw read BAMs. (default: %(default)s)

**coverage_only**

```
usage: reports.py coverage_only [-h]
                                [--cov_thresholds COV_THRESHOLDS [COV_
↪THRESHOLDS ...]]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version]
                                mapped_bams [mapped_bams ...] out_report
```

**Positional arguments:**

**mapped_bams** Aligned-to-self mapped bam files.

**out_report** Output report file.

**Options:**

**--cov_thresholds=(1, 5, 20, 100)** Genome coverage thresholds to report on. (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**alignment_summary**

Write or print pairwise alignment summary information for sequences in two FASTA files, including SNPs, ambiguous bases, and indels.

```
usage: reports.py alignment_summary [-h] [--outfileName OUTFILENAME]
                                    [--printCounts]
                                    inFastaFileOne inFastaFileTwo
```

**Positional arguments:**

| | |
|---|---|
| **inFastaFileOne** | First fasta file for an alignment |
| **inFastaFileTwo** | First fasta file for an alignment |

**Options:**

| | |
|---|---|
| **--outfileName** | Output file for counts in TSV format |
| **--printCounts=False** | Undocumented |

**consolidate_fastqc**

Consolidate multiple FASTQC reports into one.

```
usage: reports.py consolidate_fastqc [-h] inDirs [inDirs ...] outFile
```

**Positional arguments:**

| | |
|---|---|
| **inDirs** | Input FASTQC directories. |
| **outFile** | Output report file. |

**consolidate_spike_count**

Consolidate multiple spike count reports into one.

```
usage: reports.py consolidate_spike_count [-h] inDir outFile
```

**Positional arguments:**

| | |
|---|---|
| **inDir** | Input spike count directory. |
| **outFile** | Output report file. |

**plot_coverage**

Generate a coverage plot from an aligned bam file

```
usage: reports.py plot_coverage [-h] [--plotFormat] [--plotDataStyle]
                                [--plotStyle] [--plotWidth PLOT_WIDTH]
                                [--plotHeight PLOT_HEIGHT]
                                [--plotDPI PLOT_DPI] [--plotTitle PLOT_TITLE]
                                [--plotXLimits PLOT_X_LIMITS PLOT_X_LIMITS]
                                [--plotYLimits PLOT_Y_LIMITS PLOT_Y_LIMITS]
                                [-q BASE_Q_THRESHOLD] [-Q MAPPING_Q_THRESHOLD]
                                [-m MAX_COVERAGE_DEPTH]
                                [-l READ_LENGTH_THRESHOLD]
                                [--outSummary OUT_SUMMARY]
                                [--plotOnlyNonDuplicates]
                                [--loglevel
 ↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version] [--tmp_dir TMP_DIR]
                                [--tmp_dirKeep]
                                in_bam out_plot_file
```

**Positional arguments:**

| | |
|---|---|
| **in_bam** | Input reads, BAM format. |

**out_plot_file** The generated chart file

**Options:**

**--plotFormat** File format of the coverage plot. By default it is inferred from the file extension of out_plot_file, but it can be set explicitly via –plotFormat. Valid formats include: ps, eps, pdf, pgf, png, raw, rgba, svg, svgz, jpg, jpeg, tif, tiff

Possible choices: ps, eps, pdf, pgf, png, raw, rgba, svg, svgz, jpg, jpeg, tif, tiff

**--plotDataStyle=filled** The plot data display style. Valid options: filled, line, dots (default: %(default)s)

Possible choices: filled, line, dots

**--plotStyle=ggplot** The plot visual style. Valid options: seaborn-white, seaborn-pastel, seaborn-deep, seaborn-darkgrid, dark_background, seaborn-paper, grayscale, seaborn-muted, fivethirtyeight, seaborn-poster, seaborn-dark, seaborn-whitegrid, bmh, seaborn-bright, seaborn-dark-palette, ggplot, seaborn-notebook, seaborn-colorblind, classic, seaborn-talk, seaborn-ticks (default: %(default)s)

Possible choices: seaborn-white, seaborn-pastel, seaborn-deep, seaborn-darkgrid, dark_background, seaborn-paper, grayscale, seaborn-muted, fivethirtyeight, seaborn-poster, seaborn-dark, seaborn-whitegrid, bmh, seaborn-bright, seaborn-dark-palette, ggplot, seaborn-notebook, seaborn-colorblind, classic, seaborn-talk, seaborn-ticks

**--plotWidth=880** Width of the plot in pixels (default: %(default)s)

**--plotHeight=680** Width of the plot in pixels (default: %(default)s)

**--plotDPI=80.0** dots per inch for rendered output, more useful for vector modes (default: %(default)s)

**--plotTitle=Coverage Plot** The title displayed on the coverage plot (default: '%(default)s')

**--plotXLimits** Limits on the x-axis of the coverage plot; args are '<min> <max>'

**--plotYLimits** Limits on the y-axis of the coverage plot; args are '<min> <max>'

**-q** The minimum base quality threshold

**-Q** The minimum mapping quality threshold

**-m** The max coverage depth (default: %(default)s)

**-l** Read length threshold

**--outSummary** Coverage summary TSV file. Default is to write to temp.

**--plotOnlyNonDuplicates=False** Plot only non-duplicates (samtools -F 1024), coverage counted by bedtools rather than samtools.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**align_and_plot_coverage**

Take reads, align to reference with BWA-MEM, and generate a coverage plot

```
usage: reports.py align_and_plot_coverage [-h] [--plotFormat]
                                          [--plotDataStyle] [--plotStyle]
                                          [--plotWidth PLOT_WIDTH]
                                          [--plotHeight PLOT_HEIGHT]
                                          [--plotDPI PLOT_DPI]
                                          [--plotTitle PLOT_TITLE]
                                          [--plotXLimits PLOT_X_LIMITS PLOT_X_
↪LIMITS]
                                          [--plotYLimits PLOT_Y_LIMITS PLOT_Y_
↪LIMITS]
                                          [-q BASE_Q_THRESHOLD]
                                          [-Q MAPPING_Q_THRESHOLD]
                                          [-m MAX_COVERAGE_DEPTH]
                                          [-l READ_LENGTH_THRESHOLD]
                                          [--outSummary OUT_SUMMARY]
                                          [--outBam OUT_BAM] [--sensitive]
                                          [--excludeDuplicates]
                                          [--JVMmemory JVMMEMORY]
                                          [--picardOptions [PICARDOPTIONS␣
↪[PICARDOPTIONS ...]]]
                                          [--minScoreToFilter MIN_SCORE_TO_
↪FILTER]
                                          [--aligner {novoalign,bwa}]
                                          [--aligner_options ALIGNER_OPTIONS]
                                          [--NOVOALIGN_LICENSE_PATH NOVOALIGN_
↪LICENSE_PATH]
                                          [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                          [--version] [--tmp_dir TMP_DIR]
                                          [--tmp_dirKeep]
                                          in_bam out_plot_file ref_fasta
```

**Positional arguments:**

**in_bam** Input reads, BAM format.

**out_plot_file** The generated chart file

**ref_fasta** Reference genome, FASTA format.

**Options:**

**--plotFormat** File format of the coverage plot. By default it is inferred from the file extension of out_plot_file, but it can be set explicitly via –plotFormat. Valid formats include: ps, eps, pdf, pgf, png, raw, rgba, svg, svgz, jpg, jpeg, tif, tiff

Possible choices: ps, eps, pdf, pgf, png, raw, rgba, svg, svgz, jpg, jpeg, tif, tiff

**--plotDataStyle=filled** The plot data display style. Valid options: filled, line, dots (default: %(default)s)

Possible choices: filled, line, dots

**--plotStyle=ggplot** The plot visual style. Valid options: seaborn-white, seaborn-pastel, seaborn-deep, seaborn-darkgrid, dark_background, seaborn-paper, grayscale, seaborn-muted, fivethirtyeight, seaborn-poster, seaborn-dark, seaborn-whitegrid, bmh, seaborn-bright, seaborn-dark-palette, ggplot, seaborn-notebook, seaborn-colorblind, classic, seaborn-talk, seaborn-ticks (default: %(default)s)

Possible choices: seaborn-white, seaborn-pastel, seaborn-deep, seaborn-darkgrid, dark_background, seaborn-paper, grayscale, seaborn-muted, fivethirtyeight, seaborn-poster, seaborn-dark, seaborn-whitegrid, bmh, seaborn-bright, seaborn-dark-palette, ggplot, seaborn-notebook, seaborn-colorblind, classic, seaborn-talk, seaborn-ticks

**--plotWidth=880** Width of the plot in pixels (default: %(default)s)

**--plotHeight=680** Width of the plot in pixels (default: %(default)s)

**--plotDPI=80.0** dots per inch for rendered output, more useful for vector modes (default: %(default)s)

**--plotTitle=Coverage Plot** The title displayed on the coverage plot (default: '%(default)s')

**--plotXLimits** Limits on the x-axis of the coverage plot; args are '<min> <max>'

**--plotYLimits** Limits on the y-axis of the coverage plot; args are '<min> <max>'

**-q** The minimum base quality threshold

**-Q** The minimum mapping quality threshold

**-m** The max coverage depth (default: %(default)s)

**-l** Read length threshold

**--outSummary** Coverage summary TSV file. Default is to write to temp.

**--outBam** Output aligned, indexed BAM file. Default is to write to temp.

**--sensitive=False** Equivalent to giving bwa: '-k 12 -A 1 -B 1 -O 1 -E 1'. Only relevant if the bwa aligner is selected (the default).

**--excludeDuplicates=False** MarkDuplicates with Picard and only plot non-duplicates

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--picardOptions=[]** Optional arguments to Picard's MarkDuplicates, OPTIONNAME=value ...

**--minScoreToFilter** Filter bwa alignments using this value as the minimum allowed alignment score. Specifically, sum the alignment scores across

all alignments for each query (including reads in a pair, supplementary and secondary alignments) and then only include, in the output, queries whose summed alignment score is at least this value. This is only applied when the aligner is 'bwa'. The filtering on a summed alignment score is sensible for reads in a pair and supplementary alignments, but may not be reasonable if bwa outputs secondary alignments (i.e., if '-a' is in the aligner options). (default: not set - i.e., do not filter bwa's output)

**--aligner=bwa** aligner (default: %(default)s)

Possible choices: novoalign, bwa

**--aligner_options** aligner options (default for novoalign: "-r Random -l 40 -g 40 -x 20 -t 100 -k", bwa: bwa defaults

**--NOVOALIGN_LICENSE_PATH** A path to the novoalign.lic file. This overrides the NOVOALIGN_LICENSE_PATH environment variable. (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**fastqc**

```
usage: reports.py fastqc [-h] inBam outHtml
```

**Positional arguments:**

**inBam** Input reads, BAM format.

**outHtml** Output report, HTML format.

### 1.3.9 illumina.py - for raw Illumina outputs

Utilities for demultiplexing Illumina data.

```
usage: illumina.py subcommand
```

**Sub-commands:**

**illumina_demux**

Read Illumina runs & produce BAM files, demultiplexing to one bam per sample, or for simplex runs, a single bam will be produced bearing the flowcell ID. Wraps together Picard's ExtractBarcodes (for multiplexed samples) and IlluminaBasecallsToSam while handling the various required input formats. Also can read Illumina BCL directories, tar.gz BCL directories.

```
usage: illumina.py illumina_demux [-h] [--outMetrics OUTMETRICS]
                                  [--commonBarcodes COMMONBARCODES]
                                  [--sampleSheet SAMPLESHEET]
                                  [--runInfo RUNINFO] [--flowcell FLOWCELL]
```

```
                                    [--read_structure READ_STRUCTURE]
                                    [--max_mismatches MAX_MISMATCHES]
                                    [--minimum_base_quality MINIMUM_BASE_
↪QUALITY]
                                    [--min_mismatch_delta MIN_MISMATCH_DELTA]
                                    [--max_no_calls MAX_NO_CALLS]
                                    [--minimum_quality MINIMUM_QUALITY]
                                    [--compress_outputs COMPRESS_OUTPUTS]
                                    [--sequencing_center SEQUENCING_CENTER]
                                    [--adapters_to_check [ADAPTERS_TO_CHECK␣
↪[ADAPTERS_TO_CHECK ...]]]
                                    [--platform PLATFORM]
                                    [--max_reads_in_ram_per_tile MAX_READS_IN_
↪RAM_PER_TILE]
                                    [--max_records_in_ram MAX_RECORDS_IN_RAM]
                                    [--apply_eamss_filter APPLY_EAMSS_FILTER]
                                    [--force_gc FORCE_GC]
                                    [--first_tile FIRST_TILE]
                                    [--tile_limit TILE_LIMIT]
                                    [--include_non_pf_reads INCLUDE_NON_PF_
↪READS]
                                    [--run_start_date RUN_START_DATE]
                                    [--read_group_id READ_GROUP_ID]
                                    [--compression_level COMPRESSION_LEVEL]
                                    [--JVMmemory JVMMEMORY] [--threads THREADS]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    inDir lane outDir
```

**Positional arguments:**

| | |
|---|---|
| **inDir** | Illumina BCL directory (or tar.gz of BCL directory). This is the top-level run directory. |
| **lane** | Lane number. |
| **outDir** | Output directory for BAM files. |

**Options:**

| | |
|---|---|
| **--outMetrics** | Output ExtractIlluminaBarcodes metrics file. Default is to dump to a temp file. |
| **--commonBarcodes** | Write a TSV report of all barcode counts, in descending order. Only applicable for read structures containing "B" |
| **--sampleSheet** | Override SampleSheet. Input tab or CSV file w/header and four named columns: barcode_name, library_name, barcode_sequence_1, barcode_sequence_2. Default is to look for a SampleSheet.csv in the inDir. |
| **--runInfo** | Override RunInfo. Input xml file. Default is to look for a RunInfo.xml file in the inDir. |
| **--flowcell** | Override flowcell ID (default: read from RunInfo.xml). |
| **--read_structure** | Override read structure (default: read from RunInfo.xml). |

**--max_mismatches=1** Picard ExtractIlluminaBarcodes MAX_MISMATCHES (default: %(default)s)

**--minimum_base_quality=10** Picard ExtractIlluminaBarcodes MINIMUM_BASE_QUALITY (default: %(default)s)

**--min_mismatch_delta** Picard ExtractIlluminaBarcodes MIN_MISMATCH_DELTA (default: %(default)s)

**--max_no_calls** Picard ExtractIlluminaBarcodes MAX_NO_CALLS (default: %(default)s)

**--minimum_quality** Picard ExtractIlluminaBarcodes MINIMUM_QUALITY (default: %(default)s)

**--compress_outputs** Picard ExtractIlluminaBarcodes COMPRESS_OUTPUTS (default: %(default)s)

**--sequencing_center** Picard IlluminaBasecallsToSam SEQUENCING_CENTER (default: %(default)s)

**--adapters_to_check=('PAIRED_END', 'NEXTERA_V1', 'NEXTERA_V2')** Picard IlluminaBasecallsToSam ADAPTERS_TO_CHECK (default: %(default)s)

**--platform** Picard IlluminaBasecallsToSam PLATFORM (default: %(default)s)

**--max_reads_in_ram_per_tile=200000** Picard IlluminaBasecallsToSam MAX_READS_IN_RAM_PER_TILE (default: %(default)s)

**--max_records_in_ram=1000000** Picard IlluminaBasecallsToSam MAX_RECORDS_IN_RAM (default: %(default)s)

**--apply_eamss_filter** Picard IlluminaBasecallsToSam APPLY_EAMSS_FILTER (default: %(default)s)

**--force_gc** Picard IlluminaBasecallsToSam FORCE_GC (default: %(default)s)

**--first_tile** Picard IlluminaBasecallsToSam FIRST_TILE (default: %(default)s)

**--tile_limit** Picard IlluminaBasecallsToSam TILE_LIMIT (default: %(default)s)

**--include_non_pf_reads=False** Picard IlluminaBasecallsToSam INCLUDE_NON_PF_READS (default: %(default)s)

**--run_start_date** Picard IlluminaBasecallsToSam RUN_START_DATE (default: %(default)s)

**--read_group_id** Picard IlluminaBasecallsToSam READ_GROUP_ID (default: %(default)s)

**--compression_level=7** Picard IlluminaBasecallsToSam COMPRESSION_LEVEL (default: %(default)s)

**--JVMmemory=7g** JVM virtual memory size (default: %(default)s)

**--threads=0** Number of threads (default: 0)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**lane_metrics**

Write out lane metrics to a tsv file.

```
usage: illumina.py lane_metrics [-h] [--read_structure READ_STRUCTURE]
                                [--JVMmemory JVMMEMORY]
                                [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                [--version] [--tmp_dir TMP_DIR]
                                [--tmp_dirKeep]
                                inDir outPrefix
```

**Positional arguments:**

**inDir** Illumina BCL directory (or tar.gz of BCL directory). This is the top-level run directory.

**outPrefix** Prefix path to the *.illumina_lane_metrics and *.illumina_phasing_metrics files.

**Options:**

**--read_structure** Override read structure (default: read from RunInfo.xml).

**--JVMmemory=8g** JVM virtual memory size (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**common_barcodes**

Extract Illumina barcodes for a run and write a TSV report of the barcode counts in descending order

```
usage: illumina.py common_barcodes [-h] [--truncateToLength TRUNCATETOLENGTH]
                                   [--omitHeader] [--includeNoise]
                                   [--outMetrics OUTMETRICS]
                                   [--sampleSheet SAMPLESHEET]
                                   [--flowcell FLOWCELL]
                                   [--read_structure READ_STRUCTURE]
                                   [--max_mismatches MAX_MISMATCHES]
                                   [--minimum_base_quality MINIMUM_BASE_
↪QUALITY]
                                   [--min_mismatch_delta MIN_MISMATCH_DELTA]
                                   [--max_no_calls MAX_NO_CALLS]
                                   [--minimum_quality MINIMUM_QUALITY]
```

```
                                    [--compress_outputs COMPRESS_OUTPUTS]
                                    [--JVMmemory JVMMEMORY]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    inDir lane outSummary
```

**Positional arguments:**

**inDir** Illumina BCL directory (or tar.gz of BCL directory). This is the top-level run directory.

**lane** Lane number.

**outSummary** Path to the summary file (.tsv format). It includes several columns: (barcode1, likely_index_name1, barcode2, likely_index_name2, count), where likely index names are either the exact match index name for the barcode sequence, or those Hamming distance of 1 away.

**Options:**

**--truncateToLength** If specified, only this number of barcodes will be returned. Useful if you only want the top N barcodes.

**--omitHeader=False** If specified, a header will not be added to the outSummary tsv file.

**--includeNoise=False** If specified, barcodes with periods (".") will be included.

**--outMetrics** Output ExtractIlluminaBarcodes metrics file. Default is to dump to a temp file.

**--sampleSheet** Override SampleSheet. Input tab or CSV file w/header and four named columns: barcode_name, library_name, barcode_sequence_1, barcode_sequence_2. Default is to look for a SampleSheet.csv in the inDir.

**--flowcell** Override flowcell ID (default: read from RunInfo.xml).

**--read_structure** Override read structure (default: read from RunInfo.xml).

**--max_mismatches=1** Picard ExtractIlluminaBarcodes MAX_MISMATCHES (default: %(default)s)

**--minimum_base_quality=10** Picard ExtractIlluminaBarcodes MINIMUM_BASE_QUALITY (default: %(default)s)

**--min_mismatch_delta** Picard ExtractIlluminaBarcodes MIN_MISMATCH_DELTA (default: %(default)s)

**--max_no_calls** Picard ExtractIlluminaBarcodes MAX_NO_CALLS (default: %(default)s)

**--minimum_quality** Picard ExtractIlluminaBarcodes MINIMUM_QUALITY (default: %(default)s)

**--compress_outputs** Picard ExtractIlluminaBarcodes COMPRESS_OUTPUTS (default: %(default)s)

**--JVMmemory=8g** JVM virtual memory size (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**guess_barcodes**

Guess the barcode value for a sample name, based on the following: - a list is made of novel barcode pairs seen in the data, but not in the picard metrics - for the sample in question, get the most abundant novel barcode pair where one of the barcodes seen in the data matches one of the barcodes in the picard metrics (partial match) - if there are no partial matches, get the most abundant novel barcode pair

Limitations: - If multiple samples share a barcode with multiple novel barcodes, disentangling them is difficult or impossible

The names of samples to guess are selected: - explicitly by name, passed via argument, OR - explicitly by read count threshold, OR - automatically (if names or count threshold are omitted) based on basic outlier detection of deviation from an assumed-balanced pool with some number of negative controls

```
usage: illumina.py guess_barcodes [-h]
                                  [--readcount_threshold READCOUNT_THRESHOLD␣
↪| --sample_names [SAMPLE_NAMES [SAMPLE_NAMES ...]]]
                                  [--outlier_threshold OUTLIER_THRESHOLD]
                                  [--expected_assigned_fraction EXPECTED_
↪ASSIGNED_FRACTION]
                                  [--number_of_negative_controls NUMBER_OF_
↪NEGATIVE_CONTROLS]
                                  [--rows_limit ROWS_LIMIT]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version] [--tmp_dir TMP_DIR]
                                  [--tmp_dirKeep]
                                  in_barcodes in_picard_metrics
                                  out_summary_tsv
```

**Positional arguments:**

**in_barcodes** The barcode counts file produced by common_barcodes.

**in_picard_metrics** The demultiplexing read metrics produced by Picard.

**out_summary_tsv** Path to the summary file (.tsv format). It includes several columns: (sample_name, expected_barcode_1, expected_barcode_2, expected_barcode_1_name, expected_barcode_2_name, expected_barcodes_read_count, guessed_barcode_1, guessed_barcode_2, guessed_barcode_1_name, guessed_barcode_2_name, guessed_barcodes_read_count, match_type), where the expected values are those used by Picard during demultiplexing and the guessed values are based on the barcodes seen among the data.

**Options:**

**--readcount_threshold** If specified, guess barcodes for samples with fewer than this many reads.

**--sample_names** If specified, only guess barcodes for these sample names.

**--outlier_threshold=0.675** threshold of how far from unbalanced a sample must be to be considered an outlier.

**--expected_assigned_fraction=0.7** The fraction of reads expected to be assigned. An exception is raised if fewer than this fraction are assigned.

**--number_of_negative_controls=1** The number of negative controls in the pool, for calculating expected number of reads in the rest of the pool.

**--rows_limit=1000** The number of rows to use from the in_barcodes.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**miseq_fastq_to_bam**

Convert fastq read files to a single bam file. Fastq file names must conform to patterns emitted by Miseq machines. Sample metadata must be provided in a SampleSheet.csv that corresponds to the fastq filename. Specifically, the _S##_ index in the fastq file name will be used to find the corresponding row in the SampleSheet

```
usage: illumina.py miseq_fastq_to_bam [-h] [--inFastq2 INFASTQ2]
                                      [--runInfo RUNINFO]
                                      [--sequencing_center SEQUENCING_CENTER]
                                      [--JVMmemory JVMMEMORY]
                                      [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                      [--version] [--tmp_dir TMP_DIR]
                                      [--tmp_dirKeep]
                                     outBam sampleSheet inFastq1
```

**Positional arguments:**

**outBam** Output BAM file.

**sampleSheet** Input SampleSheet.csv file.

**inFastq1** Input fastq file; 1st end of paired-end reads if paired.

**Options:**

**--inFastq2** Input fastq file; 2nd end of paired-end reads.

**--runInfo** Input RunInfo.xml file.

**--sequencing_center** Name of your sequencing center (default is the sequencing machine ID from the RunInfo.xml)

**--JVMmemory=2g** JVM virtual memory size (default: %(default)s)

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**extract_fc_metadata**

Extract RunInfo.xml and SampleSheet.csv from the provided Illumina directory

```
usage: illumina.py extract_fc_metadata [-h]
                                       [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                       [--version] [--tmp_dir TMP_DIR]
                                       [--tmp_dirKeep]
                                       flowcell outRunInfo outSampleSheet
```

**Positional arguments:**

**flowcell** Illumina directory (possibly tarball)

**outRunInfo** Output RunInfo.xml file.

**outSampleSheet** Output SampleSheet.csv file.

**Options:**

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

### 1.3.10 broad_utils.py - for data generated at the Broad Institute

Utilities for getting sequences out of the Broad walk-up sequencing pipeline. These utilities are probably not of much use outside the Broad.

```
usage: broad_utils.py subcommand
```

**Sub-commands:**

**get_bustard_dir**

Find the basecalls directory from a Picard directory

```
usage: broad_utils.py get_bustard_dir [-h]
                                      [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                      inDir
```

**Positional arguments:**

**inDir** Picard directory

**Options:**

**--loglevel=ERROR** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**get_run_date**

Find the sequencing run date from a Picard directory

```
usage: broad_utils.py get_run_date [-h]
                                   [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                   inDir
```

**Positional arguments:**

**inDir** Picard directory

**Options:**

**--loglevel=ERROR** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**get_all_names**

Get all samples

```
usage: broad_utils.py get_all_names [-h]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    {samples,libraries,runs} runfile
```

**Positional arguments:**

**type** Type of name

Possible choices: samples, libraries, runs

**runfile** File with seq run information

**Options:**

**--loglevel=ERROR** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

### 1.3.11 ncbi.py - utilities to interact with NCBI

This script contains a number of utilities for submitting our analyses to NCBI's Genbank and SRA databases, as well as retreiving records from Genbank.

```
usage: ncbi.py subcommand
```

**Sub-commands:**

**tbl_transfer**

This function takes an NCBI TBL file describing features on a genome (genes, etc) and transfers them to a new genome.

```
usage: ncbi.py tbl_transfer [-h] [--oob_clip] [--ignoreAmbigFeatureEdge]
                            [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                            [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                            [--version]
                            ref_fasta ref_tbl alt_fasta out_tbl
```

**Positional arguments:**

| | |
|---|---|
| **ref_fasta** | Input sequence of reference genome |
| **ref_tbl** | Input reference annotations (NCBI TBL format) |
| **alt_fasta** | Input sequence of new genome |
| **out_tbl** | Output file with transferred annotations |

**Options:**

| | |
|---|---|
| **--oob_clip=False** | Out of bounds feature behavior. False: drop all features that are completely or partly out of bounds True: drop all features completely out of bounds but truncate any features that are partly out of bounds |
| **--ignoreAmbigFeatureEdge=False** | Ambiguous feature behavior. False: features specified as ambiguous ("<####" or ">####") are mapped, where possible True: features specified as ambiguous ("<####" or ">####") are interpreted as exact values |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**tbl_transfer_prealigned**

This breaks out the ref and alt sequences into separate fasta files, and then creates unified files containing the reference sequence first and the alt second. Each of these unified files is then passed as a cmap to tbl_transfer_common.

This function expects to receive one fasta file containing a multialignment of a single segment/chromosome along with the respective reference sequence for that segment/chromosome. It also expects a reference containing all reference segments/chromosomes, so that the reference sequence can be identified in the input file by name. It also expects a list of reference tbl files, where each file is named according to the ID present for its corresponding sequence in the refFasta. For each non-reference sequence present in the inputFasta, two files are written: a fasta containing the segment/chromosome for the same, along with its corresponding feature table as created by tbl_transfer_common.

```
usage: ncbi.py tbl_transfer_prealigned [-h] [--oob_clip]
                                       [--ignoreAmbigFeatureEdge]
                                       [--tmp_dir TMP_DIR] [--tmp_dirKeep]
```

```
                                        [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                        [--version]
                                        inputFasta refFasta refAnnotTblFiles
                                        [refAnnotTblFiles ...] outputDir
```

**Positional arguments:**

| | |
|---|---|
| **inputFasta** | FASTA file containing input sequences, including pre-made alignments and reference sequence |
| **refFasta** | FASTA file containing the reference genome |
| **refAnnotTblFiles** | Name of the reference feature tables, each of which should have a filename comrised of [refId].tbl so they can be matched against the reference sequences |
| **outputDir** | The output directory |

**Options:**

| | |
|---|---|
| **--oob_clip=False** | Out of bounds feature behavior. False: drop all features that are completely or partly out of bounds True: drop all features completely out of bounds but truncate any features that are partly out of bounds |
| **--ignoreAmbigFeatureEdge=False** | Ambiguous feature behavior. False: features specified as ambiguous ("<####" or ">####") are mapped, where possible True: features specified as ambiguous ("<####" or ">####") are interpreted as exact values |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s]<br>Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |

**fetch_fastas**

This function downloads and saves the FASTA files from the Genbank CoreNucleotide database given a given list of accession IDs.

```
usage: ncbi.py fetch_fastas [-h] [--api_key API_KEY] [--forceOverwrite]
                            [--combinedFilePrefix COMBINEDFILEPREFIX]
                            [--fileExt FILEEXT] [--removeSeparateFiles]
                            [--chunkSize CHUNKSIZE] [--tmp_dir TMP_DIR]
                            [--tmp_dirKeep]
                            [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                            [--version]
                            emailAddress destinationDir accession_IDs
                            [accession_IDs ...]
```

**Positional arguments:**

**emailAddress** Your email address. To access Genbank databases, NCBI requires you to specify your email address with each request. In case of excessive usage of the E-utilities, NCBI will attempt to contact a user at the email address provided before blocking access. This email address should be registered with NCBI. To register an email address, simply send an email to eutilities@ncbi.nlm.nih.gov including your email address and the tool name (tool='https://github.com/broadinstitute/viral-ngs').

**destinationDir** Output directory with where .fasta and .tbl files will be saved

**accession_IDs** List of Genbank nuccore accession IDs

**Options:**

**--api_key** Your NCBI API key. If an API key is not provided, NCBI requests are limited to 3/second. If an API key is provided, requests may be submitted at a rate up to 10/second. For more information, see: https://ncbiinsights.ncbi.nlm.nih.gov/2017/11/02/new-api-keys-for-the-e-utilities/

**--forceOverwrite=False** Overwrite existing files, if present.

**--combinedFilePrefix** The prefix of the file containing the combined concatenated results returned by the list of accession IDs, in the order provided.

**--fileExt** The extension to use for the downloaded files

**--removeSeparateFiles=False** If specified, remove the individual files and leave only the combined file.

**--chunkSize=1** Causes files to be downloaded from GenBank in chunks of N accessions. Each chunk will be its own combined file, separate from any combined file created via –combinedFilePrefix (default: %(default)s). If chunkSize is unspecified and >500 accessions are provided, chunkSize will be set to 500 to adhere to the NCBI guidelines on information retreival.

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**fetch_feature_tables**

This function downloads and saves feature tables from the Genbank CoreNucleotide database given a given list of accession IDs.

```
usage: ncbi.py fetch_feature_tables [-h] [--api_key API_KEY]
                                    [--forceOverwrite]
                                    [--combinedFilePrefix COMBINEDFILEPREFIX]
                                    [--fileExt FILEEXT]
                                    [--removeSeparateFiles]
                                    [--chunkSize CHUNKSIZE]
```

```
                                    [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version]
                                    emailAddress destinationDir accession_IDs
                                    [accession_IDs ...]
```

**Positional arguments:**

**emailAddress** Your email address. To access Genbank databases, NCBI requires you to specify your email address with each request. In case of excessive usage of the E-utilities, NCBI will attempt to contact a user at the email address provided before blocking access. This email address should be registered with NCBI. To register an email address, simply send an email to eutilities@ncbi.nlm.nih.gov including your email address and the tool name (tool='https://github.com/broadinstitute/viral-ngs').

**destinationDir** Output directory with where .fasta and .tbl files will be saved

**accession_IDs** List of Genbank nuccore accession IDs

**Options:**

**--api_key** Your NCBI API key. If an API key is not provided, NCBI requests are limited to 3/second. If an API key is provided, requests may be submitted at a rate up to 10/second. For more information, see: https://ncbiinsights.ncbi.nlm.nih.gov/2017/11/02/new-api-keys-for-the-e-utilities/

**--forceOverwrite=False** Overwrite existing files, if present.

**--combinedFilePrefix** The prefix of the file containing the combined concatenated results returned by the list of accession IDs, in the order provided.

**--fileExt** The extension to use for the downloaded files

**--removeSeparateFiles=False** If specified, remove the individual files and leave only the combined file.

**--chunkSize=1** Causes files to be downloaded from GenBank in chunks of N accessions. Each chunk will be its own combined file, separate from any combined file created via –combinedFilePrefix (default: %(default)s). If chunkSize is unspecified and >500 accessions are provided, chunkSize will be set to 500 to adhere to the NCBI guidelines on information retreival.

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**fetch_genbank_records**

This function downloads and saves full flat text records from Genbank CoreNucleotide database given a given list of accession IDs.

```
usage: ncbi.py fetch_genbank_records [-h] [--api_key API_KEY]
                                     [--forceOverwrite]
                                     [--combinedFilePrefix COMBINEDFILEPREFIX]
                                     [--fileExt FILEEXT]
                                     [--removeSeparateFiles]
                                     [--chunkSize CHUNKSIZE]
                                     [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                                     [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                     [--version]
                                    emailAddress destinationDir accession_IDs
                                     [accession_IDs ...]
```

**Positional arguments:**

**emailAddress** Your email address. To access Genbank databases, NCBI requires you to specify your email address with each request. In case of excessive usage of the E-utilities, NCBI will attempt to contact a user at the email address provided before blocking access. This email address should be registered with NCBI. To register an email address, simply send an email to eutilities@ncbi.nlm.nih.gov including your email address and the tool name (tool='https://github.com/broadinstitute/viral-ngs').

**destinationDir** Output directory with where .fasta and .tbl files will be saved

**accession_IDs** List of Genbank nuccore accession IDs

**Options:**

**--api_key** Your NCBI API key. If an API key is not provided, NCBI requests are limited to 3/second. If an API key is provided, requests may be submitted at a rate up to 10/second. For more information, see: https://ncbiinsights.ncbi.nlm.nih.gov/2017/11/02/new-api-keys-for-the-e-utilities/

**--forceOverwrite=False** Overwrite existing files, if present.

**--combinedFilePrefix** The prefix of the file containing the combined concatenated results returned by the list of accession IDs, in the order provided.

**--fileExt** The extension to use for the downloaded files

**--removeSeparateFiles=False** If specified, remove the individual files and leave only the combined file.

**--chunkSize=1** Causes files to be downloaded from GenBank in chunks of N accessions. Each chunk will be its own combined file, separate from any combined file created via –combinedFilePrefix (default: %(default)s). If chunkSize is unspecified and >500 accessions are provided, chunkSize will be set to 500 to adhere to the NCBI guidelines on information retreival.

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**prep_genbank_files**

Prepare genbank submission files. Requires .fasta and .tbl files as input, as well as numerous other metadata files for the submission. Creates a directory full of files (.sqn in particular) that can be sent to GenBank.

```
usage: ncbi.py prep_genbank_files [-h] [--comment COMMENT]
                                  [--sequencing_tech SEQUENCING_TECH]
                                  [--master_source_table MASTER_SOURCE_TABLE]
                                  [--organism ORGANISM] [--mol_type MOL_TYPE]
                                  [--biosample_map BIOSAMPLE_MAP]
                                  [--coverage_table COVERAGE_TABLE]
                                  [--tmp_dir TMP_DIR] [--tmp_dirKeep]
                                  [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                  [--version]
                                  templateFile fasta_files [fasta_files ...]
                                  annotDir
```

**Positional arguments:**

**templateFile** Submission template file (.sbt) including author and contact info

**fasta_files** Input fasta files

**annotDir** Output directory with genbank submission files (.tbl files must already be there)

**Options:**

**--comment** comment field

**--sequencing_tech** sequencing technology (e.g. Illumina HiSeq 2500)

**--master_source_table** source modifier table

**--organism** species name

**--mol_type** molecule type

**--biosample_map** A file with two columns and a header: sample and BioSample. This file may refer to samples that are not included in this submission.

**--coverage_table** A genome coverage report file with a header row. The table must have at least two columns named sample and aln2self_cov_median. All other columns are ignored. Rows referring to samples not in this submission are ignored.

**--tmp_dir=/tmp** Base directory for temp files. [default: %(default)s]

**--tmp_dirKeep=False** Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure.

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

**prep_sra_table**

This is a very lazy hack that creates a basic table that can be pasted into various columns of an SRA submission spreadsheet. It probably doesn't work in all cases.

```
usage: ncbi.py prep_sra_table [-h]
                              [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                              [--version]
                              lib_fname biosampleFile md5_fname outFile
```

**Positional arguments:**

**lib_fname** A file that lists all of the library IDs that will be submitted in this batch

**biosampleFile** A file with two columns and a header: sample and BioSample. This file may refer to samples that are not included in this submission.

**md5_fname** A file with two columns and no header. Two columns are MD5 checksum and filename. Should contain an entry for every bam file being submitted in this batch. This is typical output from "md5sum *.cleaned.bam".

**outFile** Output table that contains most of the variable columns needed for SRA submission.

**Options:**

**--loglevel=INFO** Verboseness of output. [default: %(default)s]

Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION

**--version, -V** show program's version number and exit

### 1.3.12 file_utils.py - utilities to perform various file manipulations

Utilities for dealing with files.

```
usage: file_utils.py subcommand
```

**Sub-commands:**

**merge_tarballs**

Merges separate tarballs into one tarball data can be piped in and/or out

```
usage: file_utils.py merge_tarballs [-h]
                                    [--extractToDiskPath EXTRACT_TO_DISK_PATH]
                                    [--pipeInHint PIPE_HINT_IN]
                                    [--pipeOutHint PIPE_HINT_OUT]
                                    [--threads THREADS]
                                    [--loglevel
↪{DEBUG,INFO,WARNING,ERROR,CRITICAL,EXCEPTION}]
                                    [--version] [--tmp_dir TMP_DIR]
                                    [--tmp_dirKeep]
                                    out_tarball in_tarballs [in_tarballs ...]
```

**Positional arguments:**

| | |
|---|---|
| **out_tarball** | output tarball (*.tar.gz\|*.tar.lz4\|*.tar.bz2\|-); compression is inferred by the file extension. Note: if "-" is used, output will be written to stdout and –pipeOutHint must be provided to indicate compression type when compression type is not gzip (gzip is used by default). |
| **in_tarballs** | input tarballs (*.tar.gz\|*.tar.lz4\|*.tar.bz2) |

**Options:**

| | |
|---|---|
| **--extractToDiskPath** | If specified, the tar contents will also be extracted to a local directory. |
| **--pipeInHint=gz** | If specified, the compression type used is used for piped input. |
| **--pipeOutHint=gz** | If specified, the compression type used is used for piped output. |
| **--threads** | Number of threads (default: all available cores) |
| **--loglevel=INFO** | Verboseness of output. [default: %(default)s] |
| | Possible choices: DEBUG, INFO, WARNING, ERROR, CRITICAL, EXCEPTION |
| **--version, -V** | show program's version number and exit |
| **--tmp_dir=/tmp** | Base directory for temp files. [default: %(default)s] |
| **--tmp_dirKeep=False** | Keep the tmp_dir if an exception occurs while running. Default is to delete all temp files at the end, even if there's a failure. |

## 1.4 Using the WDL pipelines

Rather than chaining together viral-ngs pipeline steps as series of tool commands called in isolation, it is possible to execute them as a complete automated pipeline, from processing raw sequencer output to creating files suitable for GenBank submission. This utilizes the Workflow Description Language, which is documented at: https://github.com/openwdl/wdl

**This documentation is not yet complete**

### 1.4.1 Executing WDL workflows locally with Cromwell

See example here: https://github.com/broadinstitute/viral-ngs/blob/master/travis/tests-cromwell.sh

### 1.4.2 Executing WDL workflows on Google Cloud Platform with Cromwell

This should help: https://github.com/broadinstitute/viral-ngs/blob/master/pipes/WDL/cromwell.gcid-viral-seq.conf

### 1.4.3 Executing WDL workflows on FireCloud

More info later

### 1.4.4 Executing WDL workflows on DNAnexus

This is the primary mode of execution for many of our collaborators and lab members. You can obtain the latest versions here: https://platform.dnanexus.com/projects/F8PQ6380xf5bK0Qk0YPjB17P/data/

## 1.5 Using the Snakemake pipelines

Rather than chaining together viral-ngs pipeline steps as series of tool commands called in isolation, it is possible to execute them as a complete automated pipeline, from processing raw sequencer output to creating files suitable for GenBank submission. This utilizes Snakemake, which is documented at: https://snakemake.readthedocs.io/en/stable/

Here is an overview of the Snakemake rule graph:


download_lastal_sources
build_lastal_db
depletion
filter_to_taxon
spikein_report
merge_one_per_sample
assemble_trinity
download_reference_genome
orient_and_impute
consolidate_spike_count
refine_assembly_1
refine_assembly_2
map_reads_to_self
multi_align_mafft
fastqc_report
isnvs_per_sample
consolidate_fastqc
isnvs_vcf
all

### 1.5.1 Installation instructions

It is recommended to install the viral-ngs conda package from the `broad-viral` channel, as detailed in the installation section of this documentation.

The script `easy-deploy-viral-ngs.sh` can be used to install conda and the viral-ngs package on a standard Linux system, as well as to create new project analysis directories. Project directories can also be created manually as described below.

### 1.5.2 Setting up an analysis directory

#### Copying and creating project directories and files

The Snakemake pipline is intended to be run on an input one or more sequencer bam files, each having a filename represending a sample name. The output files are named with the same sample names, and are organized into folders corresponding to the steps of the pipeline in which they were created.

To get started, create an analysis directory somewhere in your compute environment to contain the pipeline input and output files.

Into this directory, copy the following file from the `viral-ngs/pipes` directory:

```
config.yaml
Snakefile
```

Since the file `config.yaml` is project-specific, you will need to make changes to it as approprate for your usage. The config file changes are described in greater detail below.

Next, `cd` to the analysis directory and create symbolic links to the following:

- The viral-ngs virtual environment:

  ```
  ln -s /path/to/venv-viral-ngs venv
  ```

- The viral-ngs project, checked out from GitHub or extracted from a version-tagged archive:

  ```
  ln -s /path/to/viral-ngs bin
  ```

Within the analysis directory, create the directories and files used by the Snakemake pipeline:

```
data/
    00_raw/
    01_cleaned/
    01_per_sample/
    02_align_to_self/
    02_assembly/
    03_align_to_ref/
    03_interhost/
    04_intrahost/
log/
reports/
tmp/
```

The directory structure created needs to match the locations specified in `config.yaml` .

### Adding input data

- Copy each of the raw sample bam files to the `00_raw/` directory and ensure the file names follow the convention of `{sample}.bam`.
- Create a file, `samples-depletion.txt`, to list all of the samples that should be run through the depletion pipeline, with one samplename per line as {sample}, following the format of the input bam file: `{sample}.bam`. For example, if you copied a file called "G1190.bam" into `00_raw/`, then then `samples-depletion.txt` would contain the line:

  ```
  G1190
  ```

- Create a file, `samples-assembly.txt`, to list all of the samples that should be run through the assembly pipeline.
- Create a file, `samples-runs.txt`, to list all of the samples that should be run through the int**er**host analysis pipeline.
- Create a blank file, `samples-assembly-failures.txt`, that may be filled in later.

### Modifying the `config.yaml` file

Minimal modification to the config file is necessary, though there are a few things you need to specify:

An email address for when the pipeline fetches reference data from the NCBI via their Entrez API:

```
email_point_of_contact_for_ncbi: "someone@example.com"
```

The path to the depletion databases to be used by BMTagger, along with the file prefixes of the specific databases to use. The process for creating BMTagger depletion databases is described in the NIH BMTagger docs.

```
bmtagger_db_dir: "/path/to/depletion_databases"
bmtagger_dbs_remove:
  - "hg19"
  - "GRCh37.68_ncRNA-GRCh37.68_transcripts-HS_rRNA_mitRNA"
  - "metagenomics_contaminants_v3"
```

Pre-built depletion databases are available in both *.tar.gz and *.lz4 format, for removing human reads and common metagenomic contaminants:

- GRCh37.68_ncRNA-GRCh37.68_transcripts-HS_rRNA_mitRNA.tar.gz (*.lz4)
- hg19.tar.gz (*.lz4)
- metagenomics_contaminants_v3.tar.gz (*.lz4)

Note that these databases must be extracted prior to use.

In addition to the databases used by BMTagger, you will need to specify the location and file prefix of the BLAST database to be used for depletion. The process for creating the BLAST database is described in the NIH BLAST docs, and on this website from the University of Oxford.

```
blast_db_dir: "/path/to/depletion_databases"
blast_db_remove: "metag_v3.ncRNA.mRNA.mitRNA.consensus"
```

A pre-built depletion database is also available for BLAST:

- metag_v3.ncRNA.mRNA.mitRNA.consensus.tar.gz (*.lz4)

Note that this database must be extracted prior to use.

Additional databases are needed to perform metagenomic classification using Kraken, Diamond, or Krona.

```
kraken_db: "/path/to/kraken_full_20150910"

diamond_db: "/path/to/diamond_db/nr"

krona_db: "/path/to/krona"
```

Pre-built databases for Kraken, Diamond, and Krona are available:

- kraken_ercc_db_20160718.tar.gz including ERCC spike-in RNA seqs (*.lz4)
- kraken_db.tar.gz (*.lz4)
- krona_taxonomy_20160502.tar.gz (*.lz4)
- nr.dmnd.gz (*.lz4)

Note that these databases must be extracted prior to use.

An array of the NCBI GenBank CoreNucleotide accessions for the sequences comprising the reference genome to be used for contig assembly as well as for int**er**host and int**ra**host variant analysis. In addition, you will need to specify a file prefix to be used to represent the full reference genome file used downstream.

```
accessions_for_ref_genome_build:
  - "KJ660346.2"
```

An optional file containing a list of accessions may be specified for filtering reads via LAST. This is intended to narrow to a genus. If this file is not provided, viral-ngs defaults to using the accessions specified for the reference genome.

```
accessions_file_for_lastal_db_build: "/path/to/lastal_accessions.txt"
```

A FASTA file to be used by Trimmomatic during assembly to remove contaminents from reads:

```
trim_clip_db: "/path/to/depletion_databases/contaminants.fasta"
```

Pre-built databases for Trimmomatic:

- contaminants.fasta.tar.gz (*.lz4)

A FASTA file containing spike-ins to be reported:

```
spikeins_db: "/path/to/references/ercc_spike-ins.fasta"
```

#### Modifying the `Snakefile`

Depending on the state of your input data, and where in the pipeline it may enter, it may be necessary to omit certain processing steps. For example, if your sequencing center has already demultiplexed your data and no demultiplexing is needed, you can comment out the following line in the `Snakefile` :

```
include: os.path.join(pipesDir, 'demux.rules')
```

### 1.5.3 Running the pipeline

#### Configuring for your compute platform

#### Running the pipeline directly

After the above setup is complete, run the pipeline directly by calling `snakemake` within the analysis directory.

#### When things go wrong

The pipeline may fail with errors during execution, usually while generating assemblies with Trinity. If this occurs, examine the output, add the failing sample names to `samples-assembly-failures.txt`, keeping the good ones in `samples-assembly.txt`, and re-run the pipeline. Due to sample degradation prior to sequencing in the wet lab, not all samples have the integrity to complete the pipeline, and it may necessary to skip over these samples by adding them to the `samples-assembly-failures.txt`.

### 1.5.4 Assembly of pre-filtered reads

### 1.5.5 Taxonomic filtration of raw reads

### 1.5.6 Starting from Illumina BCL directories

When starting from Illumina run directories, the viral-ngs Snakemake pipeline can demultiplex raw BCL files, and merge samples from multiple flowcell lanes or libraries. To use viral-ngs in this way, create the following files:

`flowcells.txt` (example below): A tab-delimited file describing the flowcells to demultiplex, as well as the lane to use, a path to the file listing the barcodes used in the lane, the `bustard_dir` (the run directory as written by an Illumina sequencer), and an optional column for `max_mismatches`, which specifies how many bases are allowed to differ for a read to be assigned to a particular barcode (default: 0). The column `max_mismatches` may be omitted, including its header.

```
flowcell        lane    barcode_file    bustard_dir     max_mismatches
H32G3ADXY       1       /path/to/barcodes.txt    /path/to/illumina/run/directory/run_
↪BH32G3ADXY 1
H32G3ADXY       2       /path/to/barcodes.txt    /path/to/illumina/run/directory/run_
↪BH32G3ADXY 1
AKJ6R   1       /path/to/barcodes.txt      /path/to/illumina/run/directory/run_AKJ6R ␣
↪    1
```

`barcodes.txt` (example below): A tab-delimited file describing the barcodes used for a given sample, along with a library ID.

```
sample  barcode_1       barcode_2       library_id_per_sample
41C     TAAGGCGA        TATCCTCT        AP2
21P     CGTACTAG        TATCCTCT        AP2
42C     AGGCAGAA        TATCCTCT        AP2
41P     TCCTGAGC        TATCCTCT        AP2
42P     GGACTCCT        TATCCTCT        AP2
61C     TAGGCATG        TATCCTCT        AP2
61P     CTCTCTAC        AGAGTAGA        AP2
62C     CAGAGAGG        AGAGTAGA        AP2
62P     GCTACGCT        AGAGTAGA        AP2
142C    CGAGGCTG        AGAGTAGA        AP2
WATERCTL        AAGAGGCA        AGAGTAGA        AP2
```

`samples-depletion.txt` : the list of sample names to deplete *as described above*.