

---

# **viral-ngs Documentation**

***Release v2.0.21.0***

**Broad Institute Viral Genomics**

**2020-04-30**



---

## Contents

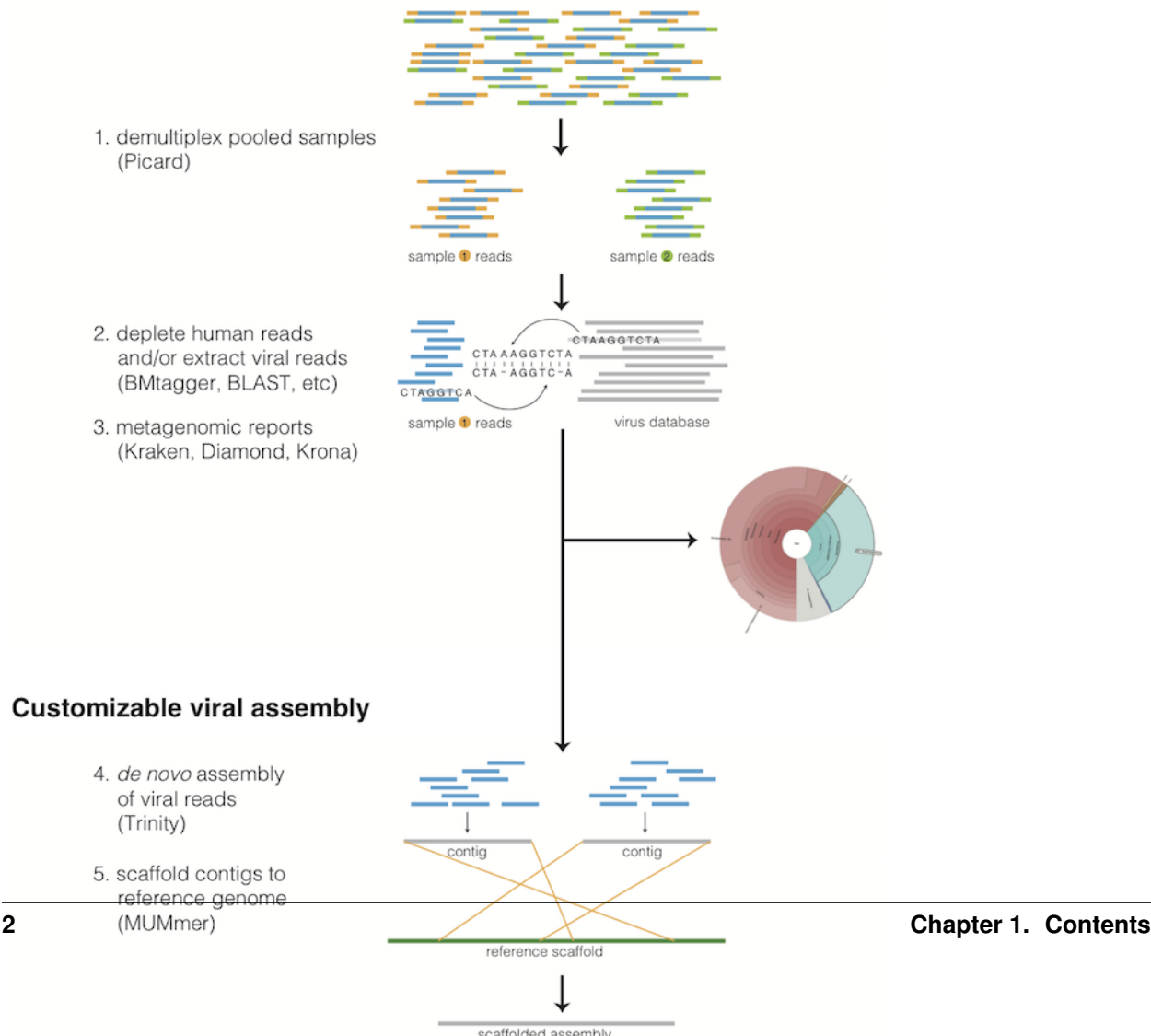
---

<b>1</b>	<b>Contents</b>	<b>1</b>
1.1	Description of the methods . . . . .	2
1.2	Using the WDL pipelines . . . . .	3
1.3	WDL Workflows . . . . .	4





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* assembly 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 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>

There are various methods for executing these workflows on your infrastructure which are more thoroughly documented in our [README](#).

## 1.3 WDL Workflows

Documentation for each workflow is provided here. Although there are many workflows that serve different functions, some of the primary workflows we use most often include:

- `demux_plus` (on every sequencing run)
- `classify_krakenuniq` (included in `demux_plus`)
- `assemble_denovo` (for most viruses)
- `assemble_refbased` (for less diverse viruses, such as those from single point source human outbreaks)
- `build_augur_tree` (for nextstrain-based visualization of phylogeny)

### 1.3.1 `align_and_count_report`

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.2 `align_and_plot`

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.3 `assemble_denovo`

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)



### 1.3.4 assemble\_denovo\_with\_deplete

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.5 assemble\_denovo\_with\_deplete\_and\_isnv\_calling

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.6 assemble\_denovo\_with\_isnv\_calling

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.7 assemble\_refbased

Reference-based microbial consensus calling. Aligns short reads to a singular reference genome, calls a new consensus sequence, and emits: new assembly, reads aligned to provided reference, reads aligned to new assembly, various figures of merit, plots, and QC metrics. The user may provide unaligned reads spread across multiple input files and this workflow will parallelize alignment per input file before merging results prior to consensus calling.

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.8 bams\_multiqc

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.9 build\_augur\_tree

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.10 classify\_kaiju

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.11 classify\_krakenuniq

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.12 contigs

#### Inputs

#### Required inputs

## Other inputs

Generated using WDL AID (0.1.1)

### 1.3.13 coverage\_table

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.14 demux\_metag

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.15 demux\_only

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.16 demux\_plus

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### **1.3.17 deplete\_only**

#### **Inputs**

#### **Required inputs**

#### **Other inputs**

Generated using WDL AID (0.1.1)

### **1.3.18 downsample**

#### **Inputs**

#### **Required inputs**

#### **Other inputs**

Generated using WDL AID (0.1.1)

### **1.3.19 fetch\_annotations**

#### **Inputs**

#### **Required inputs**

#### **Other inputs**

Generated using WDL AID (0.1.1)

### **1.3.20 multi\_Fetch\_SRA\_to\_BAM**

#### **Inputs**

#### **Required inputs**

#### **Other inputs**

Generated using WDL AID (0.1.1)

### **1.3.21 filter\_classified\_bam\_to\_taxa**

#### **Inputs**

#### **Required inputs**

## Other inputs

Generated using WDL AID (0.1.1)

### 1.3.22 genbank

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.23 isnvs\_merge\_to\_vcf

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.24 isnvs\_one\_sample

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.25 mafft

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.26 mafft\_and\_trim

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.27 merge\_bams

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.28 merge\_tar\_chunks

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.29 multiqc\_only

#### Inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.30 scaffold\_and\_refine

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)

### 1.3.31 trimal

#### Inputs

#### Required inputs

#### Other inputs

Generated using WDL AID (0.1.1)