OneRoof Pipeline File Reference

A comprehensive guide to every file in the repository

Overview

This document provides a comprehensive reference for all files in the OneRoof bioinformatics pipeline repository. Files are organized by directory to help you quickly find what you're looking for.

Root Directory Files

Core Pipeline Files

main.nf - The main entry point for the Nextflow pipeline - Orchestrates the selection and execution of platform-specific workflows (Nanopore vs Illumina) - Handles parameter validation and workflow routing - Essential for running the pipeline

nextflow.config - Central configuration file for the Nextflow pipeline - Defines default parameters, process configurations, and execution profiles - Controls resource allocation, container settings, and platform-specific behaviors - Must be understood for pipeline customization and optimization

Documentation and Configuration

README.md - Primary documentation for users - Contains installation instructions, usage examples, and quick start guides - First point of reference for new users

CLAUDE.md - AI assistant guidelines for code development - Defines project structure, key commands, and development practices - Useful for maintaining consistency in AI-assisted development

pyproject.toml - Python package configuration and dependencies - Defines project metadata, dependencies, and tool configurations - Essential for Python environment setup

pixi.lock - Lock file for Pixi environment manager - Ensures reproducible environments across different systems - Critical for dependency management

justfile - Task runner configuration (similar to Makefile) - Defines common development tasks like building Docker images and generating docs - Speeds up development workflow

Environment and Container Files

Containerfile - Docker/Podman container definition for the pipeline - Defines the execution environment with all required tools - Essential for reproducible, portable execution

flake.nix & **flake.lock** - Nix package manager configuration files - Provides an alternative reproducible environment setup - Useful for Nix users and HPC environments

uv.lock - UV package manager lock file - Alternative Python dependency management - Ensures exact Python package versions

Build and Configuration Files

_quarto.yml - Quarto documentation system configuration - Controls documentation rendering settings - Used for building the documentation website

refman.toml - Reference management configuration - May be used for managing citations or references in documentation

nf-test.config - Configuration for Nextflow testing framework - Defines test settings and locations - Important for pipeline testing and validation

data_manifest.yml - Data file manifest configuration - May define expected data structures or test data locations - Useful for data validation

llms.txt - LLM context file - Contains project information for AI assistants - Helps maintain consistent AI interactions

LICENSE - Software license file - Defines terms of use and distribution - Legal requirement for open source software

Development Files

my_script.ml & **my_script.mli** - OCaml source files (interface and implementation) - Purpose unclear - may be experimental or legacy - Could be related to performance-critical components

workflows/ Directory

Platform-specific workflow definitions that orchestrate the entire analysis pipeline:

illumina.nf - Complete workflow for processing Illumina paired-end sequencing data - Handles FASTQ input, quality control, alignment, variant calling, and consensus generation - Optimized for short-read sequencing characteristics

nanopore.nf - Complete workflow for processing Oxford Nanopore sequencing data - Supports pod5, BAM, and FASTQ inputs with optional basecalling - Handles long-read specific challenges and parameters

subworkflows/ Directory

Modular workflow components that can be reused across different main workflows:

alignment.nf - Handles read alignment to reference genomes - Integrates minimap2 with platform-specific parameters - Produces sorted, indexed BAM files for downstream analysis

consensus_calling.nf - Generates consensus sequences from aligned reads - Implements platform-specific frequency thresholds - Critical for producing final genomic sequences

gather_illumina.nf - Collects and validates Illumina FASTQ files - Handles paired-end read organization - Prepares data for processing pipeline

gather_nanopore.nf - Collects Nanopore data from various formats (pod5, BAM, FASTQ) - Handles barcode demultiplexing - Manages basecalling workflow integration

haplotyping.nf - Performs viral haplotype reconstruction - Uses Devider tool for identifying viral quasispecies - Important for studying viral diversity

illumina_correction.nf - Applies error correction specific to Illumina data - May include adapter trimming and quality filtering - Improves downstream analysis accuracy

metagenomics.nf - Performs metagenomic profiling using Sylph - Identifies organisms present in samples - Useful for contamination detection and co-infections

phylo.nf - Phylogenetic analysis using Nextclade - Assigns sequences to clades and identifies mutations - Essential for epidemiological tracking

primer_handling.nf - Manages primer validation, trimming, and analysis - Ensures complete amplicon coverage - Critical for amplicon sequencing workflows

quality_control.nf - Comprehensive quality control workflow - Integrates FastQC, MultiQC, and custom metrics - Produces quality reports for decision making

slack_alert.nf - Sends notifications to Slack channels - Reports pipeline completion status - Useful for monitoring long-running analyses

variant_calling.nf - Identifies genetic variants from aligned reads - Uses ivar for amplicon data, bcftools for general data - Produces VCF files for downstream analysis

modules/ Directory

Individual process definitions for specific bioinformatics tools:

Basecalling and Preprocessing

dorado.nf - Oxford Nanopore basecaller integration - Converts pod5 files to FASTQ with quality scores - Requires GPU for optimal performance

chopper.nf - Quality filtering for long reads - Removes low-quality Nanopore sequences - Improves downstream analysis quality

fastp.nf - Fast preprocessing for Illumina reads - Performs quality filtering and adapter trimming - Generates QC reports

cutadapt.nf - Adapter and primer trimming tool - Removes sequencing artifacts - Essential for accurate variant calling

Alignment and Coverage

minimap2.nf - Versatile sequence aligner - Handles both short and long reads - Primary alignment tool in the pipeline

samtools.nf - SAM/BAM file manipulation - Sorting, indexing, and filtering alignments - Essential for BAM file processing

mosdepth.nf - Fast coverage depth calculation - Generates coverage statistics and plots - Important for quality assessment

cramino.nf - CRAM/BAM file statistics - Provides quick alignment metrics - Useful for QC checks

Variant Calling and Consensus

ivar.nf - Variant calling and consensus for amplicon data - Handles primer trimming and frequency-based calling - Primary tool for viral genomics

bcftools.nf - General-purpose variant calling and manipulation - VCF file processing and filtering - Complementary to ivar for specific tasks

snpeff.nf - Variant annotation tool - Predicts functional effects of variants - Important for biological interpretation

Quality Control and Reporting

fastqc.nf - Sequence quality control - Generates detailed quality metrics - Standard tool for NGS QC

multiqc.nf - Aggregates QC reports from multiple tools - Creates unified quality report - Essential for multi-sample projects

plot_coverage.nf - Custom coverage visualization - Creates coverage plots per amplicon - Helps identify coverage gaps

reporting.nf - Generates analysis reports - Compiles results into readable formats - User-facing output generation

Specialized Tools

nextclade.nf - Viral clade assignment and phylogenetics - Identifies mutations and QC issues - Essential for SARS-CoV-2 and influenza analysis

sylph.nf - Metagenomic profiling - Fast organism identification - Useful for contamination detection

devider.nf - Viral haplotype reconstruction - Identifies quasispecies in samples - Important for studying viral diversity

amplicon-tk.nf - Amplicon analysis toolkit - May provide amplicon-specific utilities - Supports targeted sequencing workflows

Utility Modules

bedtools.nf - BED file manipulation - Genomic interval operations - Used for primer and region handling

seqkit.nf - Sequence manipulation toolkit - FASTA/FASTQ processing utilities - General sequence handling

rasusa.nf - Read subsampling tool - Reduces coverage to specified depth - Helps manage computational resources

vsearch.nf - Sequence clustering and searching - May be used for contamination detection - Supports sequence similarity analyses

duckdb.nf - SQL database for data analysis - Likely used for aggregating results - Enables complex data queries

grepq.nf - Pattern matching in sequences - Quick sequence searching - Utility for sequence filtering

bbmap.nf - BBMap tool suite integration - Various sequence processing utilities - Alternative/complementary to other tools

deacon.nf - Purpose unclear from name alone - May be related to decontamination - Requires investigation of module content

Pipeline-Specific Modules

validate.nf - Input validation module - Checks file formats and parameters - Ensures pipeline requirements are met

primer_patterns.nf - Generates primer search patterns - Supports primer identification in reads - Important for primer trimming

split_primer_combos.nf - Splits primers by combinations - Handles complex primer schemes - Supports multiplexed amplicons

resplice_primers.nf - Re-splices primer sequences - May handle primer artifacts - Specialized primer processing

write_primer_fasta.nf - Outputs primers in FASTA format - Utility for primer sequence export - Supports downstream analyses

output_primer_tsv.nf - Exports primer information as TSV - Creates tabular primer summaries - Useful for documentation

concat_consensus.nf - Concatenates consensus sequences - Combines multi-segment genomes - Important for segmented viruses

file_watcher.nf - Monitors directories for new files - Enables real-time processing - Supports continuous sequencing runs

call_slack_alert.nf - Sends Slack notifications - Reports pipeline events - Part of monitoring system

bin/ Directory

Python scripts and utilities for data processing:

Core Analysis Scripts

ivar_variants_to_vcf.py - Converts ivar variant output to standard VCF format - Fixes known issues with ivar's VCF generation - Essential for variant calling pipeline

plot_coverage.py - Generates coverage plots from alignment data - Creates visual representation of sequencing depth - Helps identify problematic regions

concat_consensus.py - Concatenates consensus sequences from multiple segments - Handles multi-segment viruses like influenza - Produces complete genome sequences

generate_variant_pivot.py - Creates pivot tables of variants across samples - Useful for comparing mutations between samples - Supports epidemiological analyses

Primer Management Scripts

validate_primer_bed.py - Validates primer BED file format and content - Checks for primer pair completeness - Prevents primer-related pipeline failures

make_primer_patterns.py - Generates regex patterns for primer detection - Handles primer orientation and mismatches - Supports primer trimming accuracy

split_primer_combos.py - Separates primers by pool/combination - Handles multiplexed primer schemes - Important for complex protocols

resplice_primers.py - Python implementation of primer resplicing - Handles primer artifacts in sequences - Complements Rust version

resplice_primers.rs - Rust implementation for performance - Fast primer sequence processing - Used in high-throughput scenarios

Monitoring and Utilities

file_watcher.py - Monitors directories for new sequencing files - Triggers pipeline execution automatically - Enables real-time analysis

slack_alerts.py - Sends notifications to Slack - Reports pipeline status and errors - Integrated with monitoring workflow

multisample_plot.py - Creates plots comparing multiple samples - Visualizes cross-sample metrics - Useful for batch analysis

Package Files

init.py - Python package initialization - Makes bin/ directory a Python module - Enables script imports

main.py - Package entry point - Allows running as `python -m bin` - May provide CLI interface

Test Files

test_*.py files - Unit tests for corresponding scripts - Ensures script functionality - Part of quality assurance

conf/ Directory

Configuration files for various pipeline components:

nanopore.config - Nanopore-specific pipeline settings - Defines basecalling models, parameters - Optimizes for long-read characteristics

illumina.config - Illumina-specific pipeline settings - Short-read optimized parameters - Handles paired-end specific options

snpeff.config - SnpEff variant annotation settings - Defines reference databases - Controls annotation behavior

file_watcher.template.yml - Template for file watcher configuration - Defines monitoring parameters - Customizable for different setups

assets/ Directory

Reference files and test data:

SARS-CoV-2 References

MN908947.3.fasta - SARS-CoV-2 reference genome sequence - Wuhan-Hu-1 isolate standard reference - Used for alignment and variant calling

MN908947.3.gbk - GenBank format with annotations - Contains gene and feature information - Used for variant annotation

MN908947.3_corrected_orf1.gff - Corrected ORF1 annotations - Fixes known annotation issues - Improves variant interpretation

Custom References

custom_reference.fasta - User-definable reference sequence - Supports non-standard organisms - Flexible pipeline application

annotation-custom.gbk - Custom annotation file - Pairs with custom references - Enables diverse analyses

Primer Schemes

qiaseq_direct_boosted.bed - QIAseq SARS-CoV-2 primer scheme - Commercial primer set definition - Supported primer option

final_truth_no_dashes.bed - Validated primer scheme - May be a reference standard - Used for testing/validation

Other References

h5_cattle_genome_root_segments.fasta - H5N1 influenza reference segments - Cattle-adapted strain reference - Supports influenza surveillance

lib/ Directory

Groovy libraries for Nextflow:

Utils.groovy - Utility functions for Nextflow workflows - Common functionality across workflows - Reduces code duplication

docs/ Directory

Project documentation sources:

Core Documentation

index.qmd - Main documentation page source - Renders to HTML/PDF documentation - User-facing pipeline guide

developer.qmd & **developer.md** - Developer documentation - Technical details for contributors - Code structure and patterns

pipeline_architecture.qmd & **pipeline_architecture.md** - Detailed pipeline design documentation - Architectural decisions and flow - Technical reference

data_management.qmd & **data_management.md** - Data handling guidelines - Storage and organization practices - Best practices documentation

Generated Files

pipeline_architecture_files/ - Quarto-generated web assets - JavaScript, CSS, and fonts - Supports interactive documentation

globus/ Directory

Globus integration for data transfer:

README.md - Globus setup instructions - Configuration guidelines - Integration documentation

action_provider/ - Globus action provider implementation - Enables automated workflows - Cloud integration support

config/ - Globus configuration files - Service settings - Authentication setup

flows/ - Globus flow definitions - Automated data workflows - Pipeline integration

scripts/ - Deployment and testing scripts - Globus service management - Operational utilities

tests/ Directory

Test files and data:

README.md - Test documentation - Running test instructions - Test data descriptions

data/ - Test datasets - Example files for each data type - Validation datasets

modules/, **subworkflows/**, **workflows/** - Nextflow test definitions - Unit and integration tests - Pipeline validation

GitHub Workflows (.github/)

workflows/test.yml - CI/CD test workflow - Automated testing on commits - Quality assurance

workflows/docker-image.yaml - Docker image building workflow - Automated container updates - Deployment automation

Generated/Temporary Files

These files are typically excluded from tracking:

python_respliced.bed & **rust_respliced.bed** - Output from resplicing scripts - Comparison/testing artifacts - May be temporary

test.vcf & **test_all_hap.vcf** - Test VCF outputs - Validation artifacts - Usually temporary

oneroof.egg-info/ - Python package build artifacts - Generated during installation - Not tracked in git

Summary

The OneRoof pipeline repository is organized into logical directories that separate:

1. **Core pipeline logic** (workflows/, subworkflows/, modules/)
2. **Utility scripts** (bin/)
3. **Configuration** (conf/, *.config)
4. **Documentation** (docs/, *.md)
5. **Test infrastructure** (tests/)
6. **Reference data** (assets/)
7. **External integrations** (globus/)

This structure promotes modularity, reusability, and maintainability while supporting both Nanopore and Illumina sequencing platforms for viral genomics applications.