Getting Started with the Metagenomic Pipeline

This guide will help you quickly set up and run the complete metagenomic analysis pipeline.

🚀 Quick Start (5 Minutes)

Prerequisites

- Linux system (Ubuntu 20.04+ recommended)
- 64 GB RAM minimum (128 GB+ recommended)
- 500 GB disk space for databases
- Docker, Singularity, or Conda installed
- Internet connection for database downloads

Installation

```bash
# 1. Install Nextflow
curl -s https://get.nextflow.io | bash
sudo mv nextflow /usr/local/bin/

# 2. Clone this repository
git clone https://github.com/yourusername/metagenomics-pipeline.git
cd metagenomics-pipeline

# 3. Make scripts executable
chmod +x scripts/*.sh

# 4. Verify installation
./scripts/validate_installation.sh
```

First Run (Test Data)

```bash
```

```
# Run with test dataset
nextflow run main.nf \
    --input test/test_samplesheet.csv \
    --outdir test_results \
    --skip_assembly \
    --skip_binning \
    --skip_growth_rates \
    --metaphlan_db ~/databases/metaphlan_db \
    -profile docker
```

📊 Your First Real Analysis

Step 1: Prepare Your Data

Create a samplesheet CSV file:

```
csv

sample,fastq_1,fastq_2
sample1,/path/to/sample1_R1.fastq.gz,/path/to/sample1_R2.fastq.gz
sample2,/path/to/sample2_R1.fastq.gz,/path/to/sample2_R2.fastq.gz
```

Or use the helper script:

```
bash

python scripts/generate_samplesheet.py \
    --directory /path/to/fastq_files \
    --output my_samples.csv
```

Step 2: Download Databases

```
bash

# Create database directory
mkdir -p ~/metagenomics_databases
cd ~/metagenomics_databases

# Download essential databases (this takes time!)
./scripts/setup_databases.sh

# OR download manually following SETUP.md instructions
```

Essential databases (~60 GB total):

- Human genome (for host removal): ~3 GB
- MetaPhlAn (taxonomy): ~5 GB
- HUMAnN ChocoPhlAn + UniRef (function): ~40 GB
- CheckM (bin quality): ~275 MB

Step 3: Run the Pipeline

Option A: Using the quick-start script (recommended)

```bash
./scripts/run_pipeline.sh \
    -i my_samples.csv \
    -o results \
    -d ~/metagenomics_databases \
    -p docker
```

Option B: Direct Nextflow command

```bash
nextflow run main.nf \
    --input my_samples.csv \
    --outdir results \
    --host_genome ~/metagenomics_databases/human_genome/human_GRCh38 \
    --metaphlan_db ~/metagenomics_databases/metaphlan_db \
    --humann_nucleotide_db ~/metagenomics_databases/humann_dbs/chocophlan \
    --humann_protein_db ~/metagenomics_databases/humann_dbs/uniref \
    --checkm_db ~/metagenomics_databases/checkm_data \
    -profile docker \
    -resume
```

🎯 Common Use Cases

Use Case 1: Taxonomy and Function Only (Fast)

```bash
```

```
nextflow run main.nf \
    --input samples.csv \
    --outdir results_tax_func \
    --metaphlan_db ~/databases/metaphlan_db \
    --humann_nucleotide_db ~/databases/humann_dbs/chocophlan \
    --humann_protein_db ~/databases/humann_dbs/uniref \
    --skip_assembly \
    --skip_binning \
    --skip_growth_rates \
    -profile docker
```

Time estimate: 2-4 hours for 10 samples

Output: Taxonomic profiles, functional profiles

Use Case 2: Complete MAG Recovery

```bash
nextflow run main.nf \
    --input samples.csv \
    --outdir results_mags \
    --host_genome ~/databases/human_genome/human_GRCh38 \
    --checkm_db ~/databases/checkm_data \
    --coassembly \
    --binning_tools metabat2,maxbin2,concoct \
    --min_bin_completeness 50 \
    --max_bin_contamination 10 \
    --skip_functional \
    -profile docker
```

Time estimate: 24-48 hours for 10 samples

Output: Quality-filtered MAGs (bins)

Use Case 3: Growth Rate Analysis

```bash
nextflow run main.nf \
    --input samples.csv \
    --outdir results_growth \
    --host_genome ~/databases/human_genome/human_GRCh38 \
    --checkm_db ~/databases/checkm_data \
    --coassembly \
    --binning_tools metabat2,maxbin2 \
```

```
    -profile docker
```

Time estimate: 36-72 hours for 10 samples

Output: Bacterial growth rates

Use Case 4: Full Pipeline (Everything)

```
bash

nextflow run main.nf \
    --input samples.csv \
    --outdir results_complete \
    --host_genome ~/databases/human_genome/human_GRCh38 \
    --metaphlan_db ~/databases/metaphlan_db \
    --humann_nucleotide_db ~/databases/humann_dbs/chocophlan \
    --humann_protein_db ~/databases/humann_dbs/uniref \
    --checkm_db ~/databases/checkm_data \
    --kegg_db ~/databases/kegg_db/kegg_db.dmnd \
    --cazy_db ~/databases/cazy_db/cazy_db.dmnd \
    --binning_tools metabat2,maxbin2 \
    -profile docker \
    -resume
```

Time estimate: 48-96 hours for 10 samples

Output: Everything!

🔧 Platform Setup

Docker (Local/Cloud)

```
bash

# Install Docker
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker $USER
newgrp docker

# Test
docker run hello-world
```

Singularity (HPC)

```
bash
```

```
# Usually pre-installed on HPC
module load singularity

# Or install following SETUP.md
```

SLURM (HPC Cluster)

Create a submission script:

```bash
#!/bin/bash
#SBATCH --job-name=metagenomics
#SBATCH --time=72:00:00
#SBATCH --cpus-per-task=4
#SBATCH --mem=16G
#SBATCH --output=pipeline_%j.log

module load singularity nextflow

nextflow run /path/to/metagenomics-pipeline/main.nf \
    --input samples.csv \
    --outdir results \
    --host_genome /data/databases/human_genome/human_GRCh38 \
    --metaphlan_db /data/databases/metaphlan_db \
    --humann_nucleotide_db /data/databases/humann_dbs/chocophlan \
    --humann_protein_db /data/databases/humann_dbs/uniref \
    -profile slurm \
    -resume
```

Submit:

```bash
sbatch run_pipeline.sh
```

AWS Batch (Cloud)

```bash
```

```
# 1. Upload data to S3
aws s3 sync ./data s3://my-bucket/data/
aws s3 cp samplesheet.csv s3://my-bucket/

# 2. Run pipeline
nextflow run main.nf \
    --input s3://my-bucket/samplesheet.csv \
    --outdir s3://my-bucket/results \
    --metaphlan_db s3://my-bucket/databases/metaphlan_db \
    --humann_nucleotide_db s3://my-bucket/databases/humann_dbs/chocophlan \
    --humann_protein_db s3://my-bucket/databases/humann_dbs/uniref \
    -profile awsbatch \
    -work-dir s3://my-bucket/work
```

📁 Understanding the Output

After completion, your results directory will contain:

```
results/
├── 01_kneaddata/            # Cleaned reads
├── 02_taxonomy/             # Taxonomic profiles (MetaPhlAn)
│   └── metaphlan/
│       └── sample1/
│           └── sample1_metaphlan_profile.txt
├── 03_functional/           # Functional profiles (HUMAnN)
│   └── humann/
│       └── sample1/
│           ├── sample1_genefamilies.tsv
│           ├── sample1_pathabundance.tsv
│           └── sample1_pathcoverage.tsv
├── 04_assembly/             # Assembled contigs
├── 09_binning/              # MAGs (bins)
│   ├── metabat2/
│   ├── maxbin2/
│   ├── dastool/             # Integrated bins
│   └── checkm/              # Quality reports
├── 10_growth_rates/         # Growth rate estimates
├── qc/multiqc/              # Quality control summary
│   └── multiqc_report.html
```

```
└── multiqc_report.html
└── pipeline_info/          # Execution information
    ├── execution_report.html
    └── execution_timeline.html
```

Key files to check:

- `qc/multiqc/multiqc_report.html` - Quality control overview
- `02_taxonomy/metaphlan/*/metaphlan_profile.txt` - Species abundances
- `03_functional/humann/*/pathabundance.tsv` - Pathway abundances
- `09_binning/checkm/*/checkm_results.tsv` - MAG quality
- `pipeline_info/execution_report.html` - Pipeline statistics

⚡ Tips for Success

1. Start Small

- Test with 2-3 samples first
- Use `--skip_assembly` for faster initial results
- Check quality control reports before full analysis

2. Resume Failed Runs

```bash
# Always add -resume to continue from last successful step
nextflow run main.nf [options] -resume
```

3. Monitor Progress

```bash
# Check Nextflow log
tail -f .nextflow.log

# Monitor resource usage
htop  # or top
```

4. Adjust Resources

```bash
# If running out of memory
```

```
--max_memory 256.GB

# If processes timeout
--max_time 480.h

# Limit parallel processes
--max_cpus 8
```

5. Clean Up

```
bash

# Remove work directory after successful completion
rm -rf work/

# Or use auto-cleanup
nextflow run main.nf [options] -with-cleanup
```

🪲 Troubleshooting

Problem: Out of Memory

```
bash

# Increase memory or skip assembly
nextflow run main.nf --max_memory 256.GB --skip_assembly [other options]
```

Problem: Database Not Found

```
bash

# Check database paths exist
ls ~/databases/metaphlan_db
ls ~/databases/humann_dbs/chocophlan

# Re-download if needed
cd ~/databases
metaphlan --install --bowtie2db metaphlan_db
```

Problem: Pipeline Hangs

```
bash

# Check log for stuck processes
```

```
tail -f .nextflow.log

# Kill and restart with -resume
Ctrl+C
nextflow run main.nf [options] -resume
```

Problem: Permission Denied

```
bash

# For Docker
sudo usermod -aG docker $USER
newgrp docker

# For files
chmod -R 755 /path/to/pipeline
```

📚 Next Steps

1. **Read the full documentation:**
   - `README.md` - Complete feature list
   - `SETUP.md` - Detailed installation
   - `EXAMPLES.md` - More use cases
2. **Explore your results:**
   - Start with MultiQC report
   - Analyze taxonomic profiles
   - Examine functional pathways
3. **Customize the pipeline:**
   - Adjust parameters for your study
   - Enable/disable specific analyses
   - Optimize for your compute environment
4. **Get help:**
   - Check `docs/troubleshooting.md`
   - Open GitHub issues
   - Email: ashoks773@gmail.com

🎓 Learning Resources

- **Nextflow:** https://www.nextflow.io/docs/latest/
- **MetaPhlAn:** https://huttenhower.sph.harvard.edu/metaphlan/
- **HUMAnN:** https://huttenhower.sph.harvard.edu/humann/
- **MEGAHIT:** https://github.com/voutcn/megahit
- **CheckM:** https://github.com/Ecogenomics/CheckM

✅ Checklist

Before running the pipeline, ensure:

- [ ] Nextflow is installed and working
- [ ] Docker/Singularity/Conda is configured
- [ ] Databases are downloaded
- [ ] Samplesheet is correctly formatted
- [ ] Sufficient disk space available
- [ ] Input files exist and are accessible

You're ready to start! Good luck with your metagenomic analysis! 🧬🔬