**Dorado Basecalling Guide: Converting Fast5 to POD5 and Generating FastQ**

**Introduction**

Dorado is Oxford Nanopore Technologies' (ONT) new-generation basecaller, designed for efficient and accurate conversion of raw nanopore electrical signals (stored in fast5 or pod5 formats) into DNA or RNA sequences (fastq format). This guide will walk you through the essential steps to prepare your raw fast5 data, convert it to the more efficient pod5 format, and then use Dorado to perform the basecalling, resulting in fastq files ready for downstream analysis.

This workflow was designed to be a prerequisite for downstream NextFlow pipeline analyses for 10kb/15b amplicon data and F13L amplicon data. The associated NextFlow pipelines and how to run them are available here:

10kb 15kb amplicon sequencing: [CDCgov/ONT-Seq-analysis: ONT-Seq-analysis is a specialized pipeline designed for analyzing Oxford Nanopore Technologies (ONT) sequencing data from Mpox isolates. It performs reference-based assemblies, identifies variants, designates clades, and generates a detailed quality control (QC) report.](https://github.com/CDCgov/ONT-Seq-analysis)

F13L amplicon sequencing: [CDCgov/Mpox\_AmpSeq: Mpox\_AmpSeq is a custom nextflow-style pipeline for mpox F13L amplicon sequencing. It generates reference-based consensus sequences, variant reports, NextClade outputs including clade designation, and multiple quality control metrics.](https://github.com/CDCgov/Mpox_AmpSeq/tree/main)

**Prerequisites**

Before you begin, ensure you have the following installed and configured:

1. **Miniconda3:** A minimal installer for conda, which is used for managing software environments.
2. **Dorado:** The Dorado basecaller software.
3. **POD5 Tools:** The pod5 command-line tools for converting fast5 to pod5.

It is recommended that you install POD5 within a dedicated Conda environment if unavailable as a module.

**POD5 Install**

1. **Save the YAML content:** Save the following content into a file named New\_VW.yaml in a ~/pod5/ folder.
2. **Create the Conda environment:** Open your terminal and navigate to the directory where you saved New\_VW.yaml. Then, run the following commands:
   1. Module load miniconda3
   2. conda env create -f New\_VW.yaml conda env create -f New\_VW.yaml
      1. This command will create the VW\_conda environment with all the specified packages. This environment includes pod5 and its dependencies.
3. **Activate the environment:** Once created, you can activate it using:
   1. conda activate <name\_of\_environment> #In scripts it is VW\_conda

After these steps, your pod5 conda environment will be ready for use by the dorado\_pipeline.sh script to run pod5 commands.

**Workflow Overview**

The process involves two main stages:

1. **Fast5 to POD5 Conversion:** Raw fast5 files are converted into pod5 format. pod5 is a more compact and performant format for storing nanopore signal data, optimized for faster processing by tools like Dorado.
2. **POD5 to FastQ Basecalling:** Dorado reads the pod5 files, applies a chosen basecalling model, and outputs the DNA/RNA sequences in fastq format.

**Integrated Basecalling Workflow Script**

The shell script, dorado\_basecall\_workflow.sh, that encapsulates both the fast5 to pod5 conversion and the pod5 to fastq basecalling steps. This script is designed to be submitted to the cluster; it is not recommended to run the script with qsub.

1. **Save dorado\_basecall\_workflow.sh to directory of choice.** (Note, in both NextFlow pipelines, this script is available in assets in folder /basecalling).
2. **Make dorado\_basecall\_workflow.sh executable**
   1. chmod +x dorado\_basecall\_workflow.sh
3. **Open dorado\_basecall\_workflow.sh:** DO NOT OPEN IN NOTEPAD TO EDIT! Edit with nano is fine, but it is a lot easier to use a code editing software (MobaTextEditor, Visual Studio Code).
   1. You will need to edit **TWO PARTS OF THIS SCRIPT ONLY**
      1. **Edit:** INPUT\_FAST5\_DIR=”/full/path/to/fast5/data/output”
         1. **Ex: # directory that contains your barcode subdirectories (e.g., barcode03, barcode05).**
         2. **INPUT\_FAST5\_DIR="/scicomp/instruments-pure/23-4-631\_Nanopore-MinION-NP234947/2024/PRB-2024-MPXV-10Kb-15kb-Amplicons-JiushengDeng/PRB-5-10-2024-MPXV-10K-Amplicon-Barcoding-CladeIIb-20samples-JD/10k-20samples/20240509\_0413\_MC-113388\_ATD732\_01a289e0/fast5\_pass/"**
      2. **Edit:** OUTPUT\_BASE\_DIR="/full/path/to/data/output/directory/"
         1. **EX: # Define a base output directory for all generated files relative to the current working directory.**
         2. **OUTPUT\_BASE\_DIR="./TUTORIAL\_new/10kb15kb\_test"**
4. **After you have made these edits, you are ready to run!**
   1. qsub /path/to/saved/script/directory/dorado\_basecall\_workflow.sh
      1. As always to see the status of your run use qwatch
      2. If you want to add an email to your bash script that tells you when it is done running change the top bash script as follows:

#!/bin/bash -l

#$ -o dorado\_basecall.out

#$ -e dorado\_basecall.err

#$ -N DoradoBasecall

#$ -cwd

#$ -q gpu.q

#$ -l GPU=1

#$ -M NNN1@cdc.gov #ADD THIS LINE WITH YOUR EMAIL

#$ -m abe #ADD THIS LINE

1. Congrats, your bash script should be actively running. If you run into issues check the .err and .out logs. Most likely the error will stem from an issue in file paths.
   1. This should be the layout of your output data:
   2. In /full/path/to/data/output/directory/
      1. /dorado\_output/basecalled/
         1. Output contains a single fastq file that merged all passing fastq files into one document, used as input for demultiplexing
      2. /dorado\_output/demuxed/
         1. Output contains a single fastq files per barcode and unclassified data
         2. Output also includes a barcode summary that isn’t super informative to the user so don’t worry about it