Namibia16s Workflow Documentation

This document outlines the steps taken for data processing, basecalling, quality control, and analysis of Namibia16s sequencing data using Oxford Nanopore Technologies (ONT) and various bioinformatics tools.

---

1. Data Transfer from MinION to Local Server

1.1 Transferring Data Using `rsync`

To transfer data from the MinION laptop to the local server (`raven`), use the following `rsync` command:

```
rsync -PavH /path/to/source/data/on/laptop/... raven:/ptmp/user/folder
```

To copy files **from** `raven` to your local machine:

```
rsync -PavH raven:/ptmp/source/data/... /Users/path/to/destination/local/
```

1.2 Example Command

Here is an example command used to transfer `pod5` files:

```
rsync -PavH /Users/u_erazo/Documents/LABbook/2024/Namibia_sequencing/pod5/pod5/ raven:/ptmp/merazo/Nami
```

---

2. ONT Basecalling, Filtering, and Demultiplexing

2.1 Installation of Dorado on HPC

1. Create a directory for Dorado:

```
mkdir dorado
cd dorado
```

2. Download the latest version of Dorado:

```
wget https://cdn.oxfordnanoportal.com/software/analysis/dorado-0.9.1-linux-x64.tar.gz
```

3. Extract and rename the directory:

```
tar -xf dorado-0.9.1-linux-x64.tar.gz
mv dorado-0.9.1-linux-x64 0.9.1
```

4. Verify the installation:

```
/u/merazo/dorado/0.9.1/dorado-0.9.1-linux-x64/bin/dorado --version
```

2.2 Running the Basecalling Script

Below is the script used for basecalling, filtering, and demultiplexing:

```
#!/bin/bash -l
#SBATCH --job-name=dorado-basecall
#SBATCH --mail-type=FAIL,END
#SBATCH --mail-user=erazo@mpiib-berlin.mpg.de
#SBATCH --output=/u/merazo/data/common_logs/dorado/Namibia/dorado-basecall%x_%j.out
#SBATCH --error=/u/merazo/data/common_logs/dorado/Namibia/dorado-basecall%x_%j.err
#SBATCH --chdir=./
#SBATCH --ntasks=1
# --------------------
# GPU Commands
# ---------------------
#SBATCH --constraint="gpu"
#SBATCH --gres=gpu:a100:1
#SBATCH --cpus-per-task=18
#SBATCH --mem=125000
#SBATCH --time=24:00:00

###########################

###########################
## commands

echo "Run of guppy on MPCDF."

# load CUDA
module load cuda/11.4
# Define directories and file paths
POD5_DIR="/ptmp/merazo/Namibia16s/pod5/pod5/"
OUTPUT_DIR="/ptmp/merazo/Namibia16s/output/fastq_20240227/"
BASECALLS_BAM="/ptmp/merazo/Namibia16s/output/Namibia_16S_basecalls_ONT_20240227.bam"
SUMMARY_FILE_BEFORE="/ptmp/merazo/Namibia16s/output/summaryNamibia_beforeFiltering_20240227.txt"
SUMMARY_FILE_AFTER="/ptmp/merazo/Namibia16s/output/summaryNamibia_afterFiltering_20240227.txt"
FILTERED_BAM="/ptmp/merazo/Namibia16s/output/Namibia_16S_basecalls_afterFiltering_ONT_20240227.bam"

# Display current directory for Dorado execution
echo "Running Dorado on MPCDF."
echo "Currently in directory:" `/u/merazo/dorado/0.9.1/dorado-0.9.1-linux-x64/bin`

# Display input and output directories
echo "Preparing to run Dorado"
echo "  POD5 directory: $POD5_DIR"
echo "  Output directory: $OUTPUT_DIR"

# Run basecalling using Dorado
# Basecalling converts raw POD5 data into base sequences
echo "Running basecaller..."
srun /u/merazo/dorado/0.9.1/dorado-0.9.1-linux-x64/bin/dorado basecaller --recursive --device cuda:all
echo "Basecalling completed."

# Generate a summary report before filtering
echo "Generating pre-filtering summary..."
srun /u/merazo/dorado/0.9.1/dorado-0.9.1-linux-x64/bin/dorado summary $BASECALLS_BAM > $SUMMARY_FILE_BE
echo "Pre-filtering summary generation completed."

# Filter reads based on sequence length (between 800 and 1600 bases)
echo "Filtering reads by length..."
samtools view -h $BASECALLS_BAM | awk 'length($10) >= 800 && length($10) <= 1600 || $1 ~ /^@/' | samtool
echo "Filtering completed."

# Generate a summary report after filtering
echo "Generating post-filtering summary..."
srun /u/merazo/dorado/0.9.1/dorado-0.9.1-linux-x64/bin/dorado summary $FILTERED_BAM > $SUMMARY_FILE_AFT
```

```
echo "Post-filtering summary generation completed."

# Perform demultiplexing using Dorado
echo "Running demultiplexing..."
srun /u/merazo/dorado/0.9.1/dorado-0.9.1-linux-x64/bin/dorado demux --emit-fastq --kit-name EXP-PBC096
echo "Demultiplexing completed."

echo "All tasks completed."
echo ""
```

2.3 Submitting the Job

After saving the script, submit the job using:

```
sbatch dorado_basecall_Namibia.sbatch
```

Check the job status with:

```
squeue -u <username>
```

---

3. Quality Control

3.1 Using Samtools for Quality Control

1. Generate statistics for the .BAM file before filtering:

```
samtools stats Namibia_16S_basecalls_ONT_20240227.bam > stadistics_Namibia_16S_basecalls_ONT_20240227.tx
```

2. Generate statistics for the .BAM file after filtering:

```
samtools stats Namibia_16S_basecalls_afterFiltering_ONT_20240227.bam > stadistics_Namibia_16S_basecalls_
```

3.2 Using NanoPlot for Visualization

1. Activate the NanoPlot environment:

```
cd nanopore_QC/
conda activate envs/nanopore_qc
```

2. Generate summary plots before filtering:

```
NanoPlot --summary /ptmp/merazo/Namibia16s/output/summaryNamibia_beforeFiltering_20240227.txt --logleng
```

3. Generate summary plots after filtering:

```
NanoPlot --summary /ptmp/merazo/Namibia16s/output/summaryNamibia_afterFiltering_20240227.txt --loglength
```

4. Copy the files to the local computer:

```
rsync -PavH raven:/ptmp/merazo/Namibia16s/output/Nanoplot/beforfiltering_20250227/summary-plots-log-tran

rsync -PavH raven:/ptmp/merazo/Namibia16s/output/Nanoplot/afterfiltering_20250227/summary-plots-log-tran
```

3.3 Output Summary

- **Before Filtering:**
  - File size: 7.1 GB
  - Sequences: 3,048,109

  - Average quality: 36.0
- **After Filtering:**
  - File size: 3.5 GB
  - Sequences: 2,811,704
  - Average quality: 36.2

3.4 Counting Reads per Barcode

To count the number of reads per barcode:

```
cd /ptmp/merazo/Namibia16s/output/fastq_20240227

for file in *.fastq; do echo "$file: $(grep -c '^@' "$file") reads"; done > reads_per_barcode.txt

rsync -PavH raven:/ptmp/merazo/Namibia16s/output/fastq_20240227/reads_per_barcode.txt /Users/u_erazo/Do
```

```
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode01.fastq: 36201 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode02.fastq: 58973 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode03.fastq:
24656 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode04.fastq: 53478
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode05.fastq: 1977 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode06.fastq: 74349 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode07.fastq: 41024 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode08.fastq:
53547 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode09.fastq: 36126
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode10.fastq: 32579 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode11.fastq: 39869 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode12.fastq: 25644 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode13.fastq:
40118 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode14.fastq: 33280
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode15.fastq: 29089 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode16.fastq: 51799 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode17.fastq: 28770 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode18.fastq:
44966 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode19.fastq: 51666
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode20.fastq: 64908 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode22.fastq: 40165 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode23.fastq: 47549 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode24.fastq:
33225 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode25.fastq: 37841
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode26.fastq: 186416 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode27.fastq: 34295 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode28.fastq: 47157 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode29.fastq:
47346 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode30.fastq: 46067
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode31.fastq: 45170 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode32.fastq: 29200 reads e17a8f2887894f8d7becdbeaafbc97db
PBC096__barcode33.fastq: 40601 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode34.fastq:
29827 reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode35.fastq: 42713
reads e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode36.fastq: 52967 reads
e17a8f2887894f8d7becdbeaafbc97db14bc8e66__EXP-PBC096__barcode37.fastq: 32795 reads
```

The fastq files have an average of approximately 40,946 reads per file. Barcode26.fastq have most reads : 186,416 reads and the unclassified file 332,234 reads.

4. Emu Alignment

4.1 Standard EMU Alignment (No Modifications)

Run the standard EMU alignment with default settings:

```
#!/bin/bash -l
#SBATCH --job-name=gup-basecall
#SBATCH --mail-type=FAIL,END
#SBATCH --mail-user=erazo@mpiib-berlin.mpg.de
#SBATCH --output=/u/merazo/data/common_logs/Namibia/EMU%x_%j.out
#SBATCH --error=/u/merazo/data/common_logs/Namibia/EMU/%x_%j.err
#SBATCH --chdir=./
#SBATCH --ntasks=1
# --------------------
# GPU Commands
# --------------------
#SBATCH --constraint="gpu"
#SBATCH --gres=gpu:a100:1
#SBATCH --cpus-per-task=18
#SBATCH --mem=125000
#SBATCH --time=24:00:00

###########################

## Set Variables and conda Env

source /u/merazo/miniconda3/bin/activate EMU

###########################
## commands

cd /ptmp/merazo/Namibia16s/output/

directory="/ptmp/merazo/Namibia16s/output/fastq_20240227/"

for file in $( ls ${directory}/*.fastq )
do
        emu abundance \
            --keep-counts \
            --db /u/merazo/emu/Silva_database \
            --output-dir /ptmp/merazo/Namibia16s/output/EMU/stardard/ \
            ${file}
done
```

4.2 Modified EMU Alignment

Run EMU with modified parameters:

```
#!/bin/bash -l
#SBATCH --job-name=gup-basecall
#SBATCH --mail-type=FAIL,END
#SBATCH --mail-user=erazo@mpiib-berlin.mpg.de
#SBATCH --output=/u/merazo/data/common_logs/Namibia/EMU%x_%j.out
#SBATCH --error=/u/merazo/data/common_logs/Namibia/EMU/%x_%j.err
#SBATCH --chdir=./
#SBATCH --ntasks=1
# --------------------
# GPU Commands
# --------------------
```

```
#SBATCH --constraint="gpu"
#SBATCH --gres=gpu:a100:1
#SBATCH --cpus-per-task=18
#SBATCH --mem=125000
#SBATCH --time=24:00:00

##########################

## Set Variables and conda Env

source /u/merazo/miniconda3/bin/activate EMU

##########################
## commands

cd /ptmp/merazo/Namibia16s/output/

directory="/ptmp/merazo/Namibia16s/output/fastq_20240227/"

for file in ${directory}/*.fastq
do
    emu abundance \
        --keep-counts \
        --type map-ont \
        --min-abundance 0.000000000001 \
        --keep-read-assignment \
        --keep-files \
        --output-unclassified \
        --db /u/merazo/emu/Silva_database \
        --output-dir /ptmp/merazo/Namibia16s/output/EMU/Melanie/ \
        "$file"
done
```

4.3 Copying Outputs for Further Analysis

Copy the EMU outputs to your local machine:

```
rsync -PavH raven:/ptmp/merazo/Namibia16s/output/EMU/ /Users/u_erazo/Documents/LABbook/2024/Namibia_seq
```