Supplementary Method S1: Nanopore read quality filtering and deduplication

Quality filtering

Raw Nanopore reads were quality-filtered using **Chopper** (v 0.7; van der Valk & Blaxter, 2023), a streaming filter designed for Oxford Nanopore reads that discards reads falling below user-defined quality or length thresholds in a single pass. Chopper was applied to the per-sample FASTQ files produced after demultiplexing and barcode removal by the sequencing platform. Two parameter sets were used depending on sequencing chemistry and platform:

- **Main cohort** (40 samples; PromethION P2Solo, SQK-NBD114-24, kit14; Dorado basecalling): reads were retained if they had a mean Phred quality score 15 and a length 500 bp (`chopper -q 15 -l 500`; script: `01_preprocessing/chopper_qc_p2solo_kit14.sh`).
- **Pilot cohort** (15 samples; MinION MK1C, SQK-NBD112-24, kit12; Guppy basecalling): reads were retained if they had a mean Phred quality score 12 and a length 500 bp (`chopper -q 12 -l 500`; script: `01_preprocessing/chopper_qc_mk1c_kit12.sh`).

The lower quality threshold for the pilot cohort reflects the reduced per-read accuracy characteristic of the older kit12 chemistry relative to kit14. Read counts and mean Phred quality scores at each processing stage are summarised in Supplementary Figure S1.

Deduplication

Following quality filtering, reads were screened for duplicate read identifiers and exact sequence duplicates using a custom Python script (`analyse_fastq_deduplicate.py`; available at `01_preprocessing/`). Duplicate reads were identified as a problem after they caused errors in downstream co-assembly; the overall proportion of reads removed was low across both cohorts.

The script computed two SHA-256 hashes per read: one over the combined header, sequence, and quality string (to detect fully identical records) and one over the sequence alone (to detect reads sharing identical sequence content but differing in name or quality). Deduplication was performed by read name: the first occurrence of each read identifier was retained. Where the same read name appeared with differing sequence content — a known artefact of barcode demultiplexing bleed-through — the additional record was assigned a disambiguating suffix (`_dup{n}`) and retained as a separate entry rather than being discarded, preserving biologically distinct sequences. A per-sample report summarising the counts of exact duplicates, sequence-level duplicates, and reads with shared identifiers was written alongside each deduplicated output file.

Deduplication was run as a SLURM array job on the HPC cluster, processing three samples in parallel (`01_preprocessing/deduplicate_rename.sh`). Read counts at all three pipeline stages — raw, Chopper-filtered, and deduplicated — were computed using `seqkit stats --all` (Shen et al., 2016; script: `01_preprocessing/chopper_qc.summary.sh`).

> **Note on sample G-L-020-06 (main cohort):** The deduplicated FASTQ for this sample was stored across multiple file parts due to its large size, which produced an unexpected end-of-file error when running seqkit on the combined file. This sample is excluded from Supplementary Figure S1 (39 of 40 main-cohort samples shown) but was included in all downstream assembly and binning steps.

References

- van der Valk, T. & Blaxter, M. (2023). Chopper: quality control for Oxford Nanopore reads. *Bioinformatics*, 39(5), btad311. https://doi.org/10.1093/bioinformatics/btad311
- Shen, W., Le, S., Li, Y. & Hu, F. (2016). SeqKit: a cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PLOS ONE*, 11(10), e0163962. https://doi.org/10.1371/journal.pone.0163962