

Supplementary Method S3: Read mapping for binning depth estimation

Read mapping was performed to generate per-sample coverage depth information required by the binning tools (MetaBAT2, SemiBin2, VAMB) and to provide Nanopore reads for hybrid binning of per-sample assemblies. Three distinct mapping targets were used, corresponding to the three assembly outputs described in Supplementary Method S2. All BAM files were coordinate-sorted and indexed with samtools. Coverage depth tables for MetaBAT2 were generated using `jgi_summarize_bam_contig_depths` from the MetaBAT2 package.

Mapping for final abundance profiling in anvi'o is described separately in Supplementary Method S5.

1. Per-sample hybrid assemblies

Each sample's hybrid SPAdes assembly was used as its own mapping reference.

Illumina reads (paired-end) were aligned to each sample's assembly using **bwa mem** (`bwa_illumina_to_assembly.slurm.sh` align_nanopore_to_hybrid_slurm_pilot15.sh for the pilot cohort). The assembly was indexed with **bwa index** prior to the array job; each array task checked for the index before aligning. Output BAMs were written to `Illumina_hybrid_bam_output/`.

Nanopore reads were aligned to the same per-sample assemblies using **minimap2** with the Oxford Nanopore preset (`-ax map-ont`; `minimap2_nanopore_to_assembly.slurm.sh` for the main cohort; `align_nanopore_to_hybrid_slurm_pilot15.sh` provides the equivalent for the pilot cohort). Output BAMs were written to `bam_output_nanopore/` and `bam_output_nanopore_pilot15/` respectively.

Both mapping jobs ran as SLURM array jobs (8 CPUs, 32 GB RAM per sample). The resulting Illumina and Nanopore BAM files were used as input to per-sample MetaBAT2 binning.

2. Round 1 filtered co-assembly

Two mapping runs were performed against the filtered Round 1 co-assembly, each serving a different binning tool.

2a. bwa-mem mapping (MetaBAT2 depth)

The filtered co-assembly (`megahit_filtered_coassembly/assembly.filtered.fa`) was indexed with **bwa** (`index_coassembly.sh`). All Illumina samples were then aligned using **bwa mem** as a SLURM array job (`bwa-coassembly.sh`; array 0–121, 10 concurrent, 8 CPUs / 64 GB per sample). Output BAMs were written to `bam_output_Illumina_filt_coass/` and used to compute per-contig depth with `jgi_summarize_bam_contig_depths` for MetaBAT2.

2b. Prefixed assembly + bowtie2 mapping (SemiBin2)

SemiBin2 requires contig names to be unique and parseable across assemblies. The filtered co-assembly was therefore reformatted with a **C:** prefix on all contig headers (`01_semibin2_prefix_coassembly.slurm.sh`), and a bowtie2 index was built on this prefixed FASTA (`pref_coassembly/assembly.prefixed`). All Illumina samples were aligned to the prefixed assembly using **bowtie2** (`--very-sensitive --no-unal`; `02_remap_prefix_ass_array.slurm.sh`; array 0–121, 6 concurrent, 16 CPUs / 96 GB per sample). Output BAMs were written to `mapped_prefixed/`.

3. Round 2 rare-contig co-assembly

The same two-pass mapping strategy was applied to the Round 2 (unmapped-read) co-assembly.

bwa-mem mapping: The Round 2 assembly (`ohph_megahit_coassembly_unmapped_k27/final.contigs.fa`) was indexed with `bwa` (`0_M2_03_index_rare_megahit.sh`). All Illumina samples were aligned using `bwa mem` (`0_M2_04_bwa-coassembly.sh`; array 0–121, 15 concurrent, 8 CPUs / 64 GB per sample), with output BAMs in `bam_output_Illumina_rare_coass/`. Per-contig depth was calculated with `jgi_summarize_bam_contig_depths`, writing `rare.asm.depth.txt` (`0_M2_05_metabat_depth_rare_coassembly.sh`).

Prefixed bowtie2 mapping: The Round 2 assembly was reformatted with a `C:` prefix and a bowtie2 index was built on the prefixed FASTA (`0_M2_08_add_prefix_for_semibin2.sh`; output: `rare_pref_coassembly/rare.assembly.prefixed.fa`). All Illumina samples were then remapped using bowtie2 (`0_M2_09_remap_to_prefixed_rare_coassembly.sh`; array 0–121, 6 concurrent, 16 CPUs / 96 GB per sample), with output BAMs in `mapped_rare_prefixed/`.

Summary

Mapping target	Tool	Purpose	Output directory
Per-sample hybrid SPAdes (Illumina)	<code>bwa mem</code>	Per-sample MetaBAT2	<code>Illumina_hybrid_bam_output/</code>
Per-sample hybrid SPAdes (Nanopore)	<code>minimap2 (map-ont)</code>	Per-sample MetaBAT2	<code>bam_output_nanopore/</code> , <code>bam_output_nanopore_pilot15/</code>
Filtered Round 1 co-assembly	<code>bwa mem</code>	MetaBAT2 depth	<code>bam_output_Illumina_filt_coass/</code>
Prefixed Round 1 co-assembly	<code>bowtie2</code>	SemiBin2	<code>mapped_prefixed/</code>
Round 2 rare co-assembly	<code>bwa mem</code>	MetaBAT2 depth	<code>bam_output_Illumina_rare_coass/</code>
Prefixed Round 2 rare co-assembly	<code>bowtie2</code>	SemiBin2	<code>mapped_rare_prefixed/</code>