

Supplementary File 2

The runtime program of purge_dups v1.2.5

```
#!/bin/bash

# Set variables
ASSEMBLY="your_assembly.fasta"
READS_BAM="aligned_reads.bam"
OUTPUT_DIR="purge_dups_output"
NUM_THREADS=32

# Create output directory
mkdir -p $OUTPUT_DIR

echo "Starting purge_dups pipeline"

# Step 1: Calculate depth statistics
echo "Step 1: Calculating depth statistics"
pbcstat $READS_BAM
calcuts PB.stat > cutoffs 2> calcuts.log

# Step 2: Split assembly sequences
echo "Step 2: Splitting assembly"
split_fa $ASSEMBLY > $OUTPUT_DIR/split.fasta

# Step 3: Minimap2 alignment
echo "Step 3: Minimap2 alignment"
minimap2 -xasm5 -t $NUM_THREADS $OUTPUT_DIR/split.fasta $ASSEMBLY | \
  gzip -c > $OUTPUT_DIR/split.self.paf.gz

# Step 4: Generate duplicate regions bed file
echo "Step 4: Generating duplicate regions"
purge_dups -2 -T cutoffs -c PB.base.cov $OUTPUT_DIR/split.self.paf.gz >
$OUTPUT_DIR/dups.bed 2> purge_dups.log

# Step 5: Get duplicate and retained contig lists
echo "Step 5: Getting duplicate and retained contig lists"
get_seqs -e $OUTPUT_DIR/dups.bed $ASSEMBLY

# Step 6: Process primary assembly only (if needed)
echo "Step 6: Processing primary assembly only"
# If there are haplotigs, you can choose to process only primary
```

```

# Extract primary contigs (assuming names start with 'pri')
grep "^>" $ASSEMBLY | grep "pri" | sed 's/>/' > $OUTPUT_DIR/primary_contigs.list

# Run purge_dups on primary assembly
purge_dups -d -T cutoffs -c PB.base.cov $OUTPUT_DIR/split.self.paf.gz \
    -l $OUTPUT_DIR/primary_contigs.list > $OUTPUT_DIR/primary_dups.bed

echo "purge_dups pipeline completed"

# Output statistics
echo "=== Purge_dups Statistics ==="
echo "Original assembly: $ASSEMBLY"
echo "Duplicates identified: $(wc -l < $OUTPUT_DIR/dups.bed)"
if [ -f "$OUTPUT_DIR/primary_dups.bed" ]; then
    echo "Primary assembly duplicates: $(wc -l <
$OUTPUT_DIR/primary_dups.bed)"
fi

```