# Core-SNP Phylogeny (Docker-only)

Workshop Handout

September 04, 2025

## Contents



## 1 Introduction

This handout shows how to start from SRA reads (or existing assemblies) and produce a clean, recombination-pruned **maximum-likelihood** tree that you can view in tools like **iTOL**, **FigTree**, or **Microreact**.

- **Reference used here:** GCF_000166455.1_ASM16645v2_genomic_REF.fna (*Vibrio cholerae* O1, strain **2010EL-1786**, Haiti 2010).
  *Note:* This is the reference I used. You can substitute a **close, high-quality** reference appropriate for your dataset (ideally same lineage) to maximize the core genome.
- **Tree building: IQ-TREE 2** with **ModelFinder + ASC** (ascertainment correction) on the **Gubbins-filtered SNP-only** core alignment.

- **Execution environment:** Everything runs in **Docker** (no conda needed). Works on Apple Silicon; keep --platform linux/amd64.
  *Why Docker here?* The relevant conda packages/channels weren't maintained for my Mac. If conda works for you, you can adapt the same steps to a conda environment instead.

### 1.1 Quick setup (variables & images)

```
# --- paths & parameters you can edit ---
REF=GCF_000166455.1_ASM16645v2_genomic_REF.fna   # put this FASTA in your
                                                  #working dir (or use your own close reference)
THREADS=8 # adjust for your CPU

# --- docker helpers (keep these) ---
PLAT="--platform linux/amd64"   # required on Apple Silicon; harmless elsewhere, I am working on a Mac :)
UIDGID="-u $(id -u):$(id -g)"   # make output files owned by you (not root)
MOUNT="-v $PWD:/data -w /data" # mount the current folder as /data inside containers

# --- docker images used --- #keep in mind some of these havent been updated in 2-3 years,
                            #running conda instead likely more updated option
IMG_SRA=staphb/sra-tools:latest
IMG_SHOVILL=staphb/shovill:latest
IMG_SNIPPY=staphb/snippy:latest
IMG_GUBBINS=sangerpathogens/gubbins:latest
IMG_IQTREE=nanozoo/iqtree:2.2.0.3--ba94576
IMG_SNPDISTS=staphb/snp-dists:latest
```

Notes
- I ran **everything in Docker** because conda packages for my Mac weren't reliable; if conda works for you, you can adapt the same steps.
- Some images are not updated frequently, I think snippy image is 2 or 3 years old
- Keep PLAT="--platform linux/amd64" on Apple Silicon; it avoids architecture mismatches.

### 1.2 Download reads from SRA (paired FASTQs)

- Create a text file of SRR accessions (one per line), then use fasterq-dump to split into R1/R2 and gzip:

```
cat > sra_list.txt <<'EOF'
SRRXXXXXXXX
SRRYYYYYYYY
# one SRR per line …
EOF

while read SRR; do
```

```
  docker run --rm $PLAT $UIDGID $MOUNT $IMG_SRA \
    bash -lc "fasterq-dump --split-files --threads $THREADS -O /data $SRR \
              && gzip -f ${SRR}_1.fastq ${SRR}_2.fastq"
done < sra_list.txt
```

#### 1.2.1 Make a simple sample sheet mapping IDs to read files (rename IDs as you like):

```
cat > samples_reads.tsv <<'EOF'
#ID            R1                    R2
Vc1786_Haiti    SRRXXXXXXXX_1.fastq.gz   SRRXXXXXXXX_2.fastq.gz
Vc1792_Haiti    SRRYYYYYYYY_1.fastq.gz   SRRYYYYYYYY_2.fastq.gz
# …
EOF
```

#### 1.2.2 Already have assemblies? Skip to 1.4.

### 1.3 Assemble with Shovill (SPAdes backend)

```
mkdir -p assemblies
awk 'NF && $1 !~ /^#/' samples_reads.tsv | while read ID R1 R2; do
  docker run --rm $PLAT $UIDGID $MOUNT $IMG_SHOVILL \
    shovill --R1 /data/$R1 --R2 /data/$R2 \
            --outdir /data/assemblies/$ID \
            --depth 100 --minlen 500 --cpus $THREADS
done
```

- Output per sample: assemblies//contigs.fa
- (Your contigs will show headers like sw=shovill-spades/1.1.0 … pilon.)

### 1.4 Call SNPs vs the reference with Snippy (contigs mode)

- Build a contigs table (ID  contigs path), generate a multi-run script with snippy-multi, then run it.
- It will process each sample and finish with snippy-core to make the core alignments.

```
: > samples_ctgs.tsv   # ID <TAB> absolute-path-to-contigs
for f in assemblies/*/contigs.fa; do
  ID=$(basename "$(dirname "$f")")
  echo -e "${ID}\t/data/$f" >> samples_ctgs.tsv
done

# generate the run script (uses /data paths inside the container)
```

```
docker run --rm $PLAT $UIDGID $MOUNT $IMG_SNIPPY \
  snippy-multi /data/samples_ctgs.tsv --ref /data/$REF --cpus $THREADS > run_snippy.sh

# execute the script (runs per-sample Snippy + snippy-core)
bash run_snippy.sh
```

- Key outputs (in working folder):
  - core.full.aln ← full core (invariant + variant sites)
  - core.aln ← SNP-only (polymorphic sites)
- We'll feed core.full.aln to Gubbins.

### 1.5 Remove recombination with Gubbin

```
docker run --rm $PLAT $UIDGID $MOUNT -w /data $IMG_GUBBINS \
  bash -lc 'run_gubbins.py --threads '"$THREADS"' --prefix gubbins_out core.full.aln'
```

- Outputs (in gubbins_out/):
  - *filtered_alignment* → full core, recombination masked
  - *filtered_polymorphic_sites* → SNP-only, recombination-masked
- Pick the SNP-only file:

```
ALN=$(ls gubbins_out/*filtered_polymorphic*fa* | head -n1)
echo "Using alignment: $ALN"
```

### 1.6 Build the maximum-likelihood tree with IQ-TREE (+ASC)

- Alignment is SNP-only → add ASC (ascertainment correction).

```
docker run --rm $PLAT $UIDGID $MOUNT -w /data $IMG_IQTREE \
  iqtree2 -s "$ALN" -m MFP+ASC -B 1000 -alrt 1000 -T AUTO --prefix snp_tree
```

- Outputs:
  - snp_tree.treefile → best ML tree (upload this to iTOL)
  - snp_tree.contree → bootstrap consensus (supports on nodes)
  - snp_tree.iqtree → detailed log (best-fit model, likelihoods)

### 1.7 (Optional) Pairwise SNP distance table

```
docker run --rm $PLAT $UIDGID $MOUNT -w /data $IMG_SNPDISTS \
  snp-dists -b "$ALN" > snp_distances.tsv
```

### 1.8 Visualize in iTOL (what to click)

- Upload snp_tree.treefile.
- Grouping view (clustering-only): Basic → Branch lengths = Ignore (turn scale bar off).
- Midpoint orientation: Basic → Rooting → Midpoint (balanced layout).
- Mirror the paper: Right-click a Bangladesh tip → Reroot here (or select both Bangladesh tips as outgroup).
- Colors/legend: Datasets → Color strips (country/source) + add a small legend.
- Supports: Basic → Branch options → show  95 only (UFboot/SH-aLRT).
- Export: PNG/PDF.

### 1.9 Minimal "already have assemblies" quick run

```
: > samples_ctgs.tsv
for f in assemblies/*/contigs.fa; do
  ID=$(basename "$(dirname "$f")"); echo -e "${ID}\t/data/$f" >> samples_ctgs.tsv
done

docker run --rm $PLAT $UIDGID $MOUNT $IMG_SNIPPY \
  snippy-multi /data/samples_ctgs.tsv --ref /data/$REF --cpus $THREADS > run_snippy.sh

bash run_snippy.sh

docker run --rm $PLAT $UIDGID $MOUNT -w /data $IMG_GUBBINS \
  bash -lc 'run_gubbins.py --threads '"$THREADS"' --prefix gubbins_out core.full.aln'

ALN=$(ls gubbins_out/*filtered_polymorphic*fa* | head -n1)

docker run --rm $PLAT $UIDGID $MOUNT -w /data $IMG_IQTREE \
  iqtree2 -s "$ALN" -m MFP+ASC -B 1000 -alrt 1000 -T AUTO --prefix snp_tree
```

### 1.10 What I did (so you can mirror it)

- Assemblies via Shovill (SPAdes); contig headers stamped shovill-spades/1.1.0 … pilon.
- Reference: GCF_000166455.1_ASM16645v2_genomic_REF.fna (Haiti 2010).
- Snippy in contigs mode via snippy-multi → snippy-core (core.full.aln, core.aln).
- Gubbins on core.full.aln; used the recomb-filtered SNP-only alignment.
- IQ-TREE 2: -m MFP+ASC -B 1000 -alrt 1000 → maximum-likelihood tree.
- iTOL: showed with/without branch lengths; midpoint & Bangladesh-rooted; colored tips by country; supports  95.

### 1.11 Common gotchas & quick fixes

- Root-owned output files? Keep -u ($id - u$) :(id -g) (already set).
- Apple Silicon warnings? Keep –platform linux/amd64.
- Gubbins error "use run_gubbins.py" → You are using it; don't call the internal gubbins binary.
- Tiny core / odd topology? Reference too distant or outlier sample → pick a closer ref or drop the outlier, rerun.
- Low supports in IQ-TREE? Use the Gubbins-filtered SNP-only alignment and +ASC.
- Display confusion? Rooting & branch-length visibility change the look, not the clades.