# Command Line Approach

GenEpi-BioTrain Virtual training 23

## Virulence Profiling of *E. coli* Pathotypes

**João Cardoso (joacar@dtu.dk)**
**Technical University of Denmark (DTU), National Food Institute**

10/03/2026

# How Can We Use Database with Command Line?

# How Can We Use Database with Command Line?

## Step 1: Create an environment

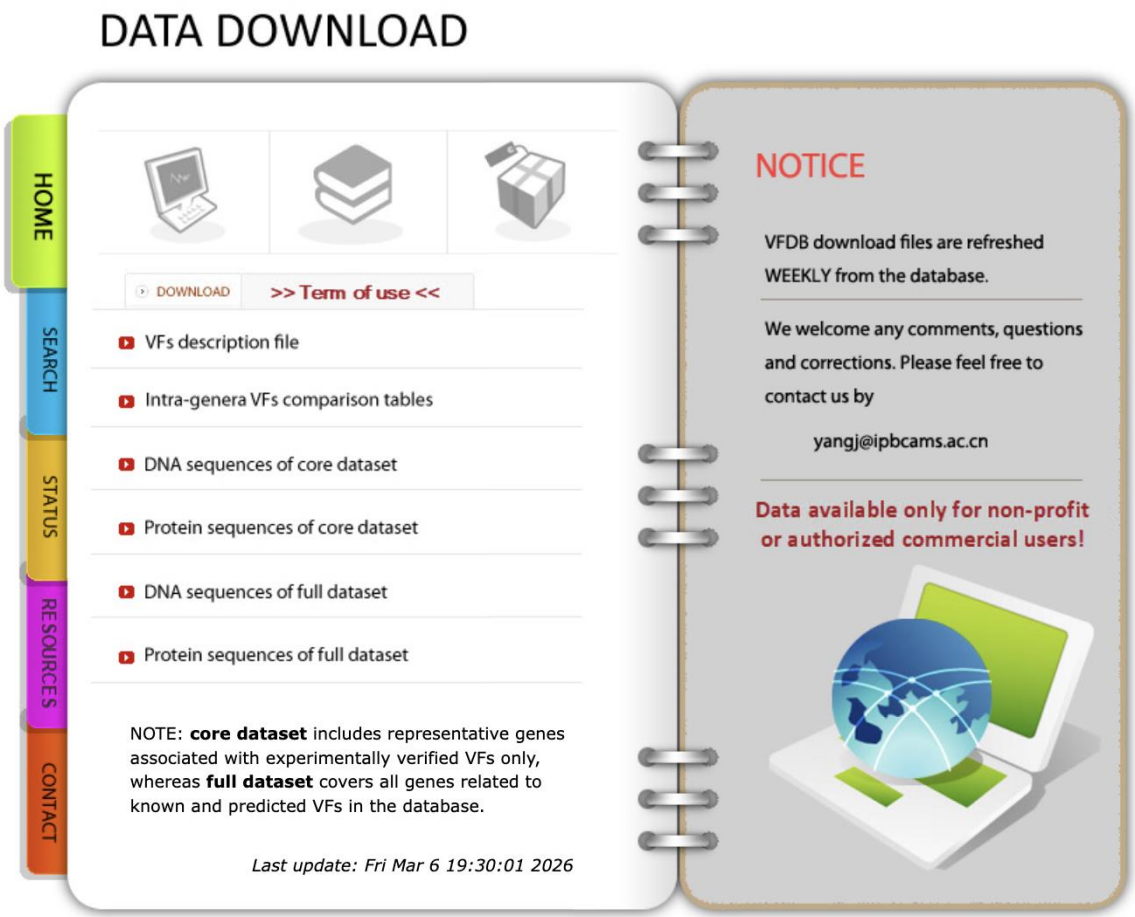
```
micromamba create -n virulence -c bioconda -c conda-forge \
    kma blast python=3.10 curl wget

micromamba activate virulence
```

## Virulence Factor Database

- Maintained at: https://www.mgc.ac.cn/VFs/
- Core set = comprehensive set (all organisms) - Curated subset of VFDB with experimentally verified virulence factors from all bacterial species.

**Protein Sequences** -> BlastX on Assemblies
**Nucleotide Sequences** -> KMA on Reads



# How Can We Use Database with Command Line?

## Step 2: Download VFDB

```
# Create a folder for the databases
mkdir -p /data/databases/Virulence/vfdb
cd /data/databases/Virulence/vfdb

# --- Protein sequences (for BLAST) ---
curl -fsSL https://www.mgc.ac.cn/VFs/Down/VFDB_setB_pro.fas.gz \
    -o vfdb_setB_pro.fas.gz
gunzip vfdb_setB_pro.fas.gz
# Result: vfdb_setB_pro.fas

# --- Nucleotide sequences (for KMA) ---
curl -fsSL https://www.mgc.ac.cn/VFs/Down/VFDB_setB_nt.fas.gz \
    -o vfdb_setB_nt.fas.gz
gunzip vfdb_setB_nt.fas.gz
# Result: vfdb_setB_nt.fas
```

## Step 3: Build the BLAST+ Database

```
# Build a BLAST protein database from VFDB proteins
makeblastdb \
    -in  vfdb_setB_pro.fas \       # Input: protein FASTA
    -dbtype prot \                  # Type: protein
    -out vfdb_setB_pro              # Output prefix (creates .phr/.pin/.psq)
```

## Step 4: Build the KMA Index

```
# Build KMA index from VFDB nucleotide sequences
kma index \
    -i vfdb_setB_nt.fas \          # Input: nucleotide FASTA
    -o vfdb_setB_nt_kma            # Output prefix
```

# How Can We Use Database with Command Line?

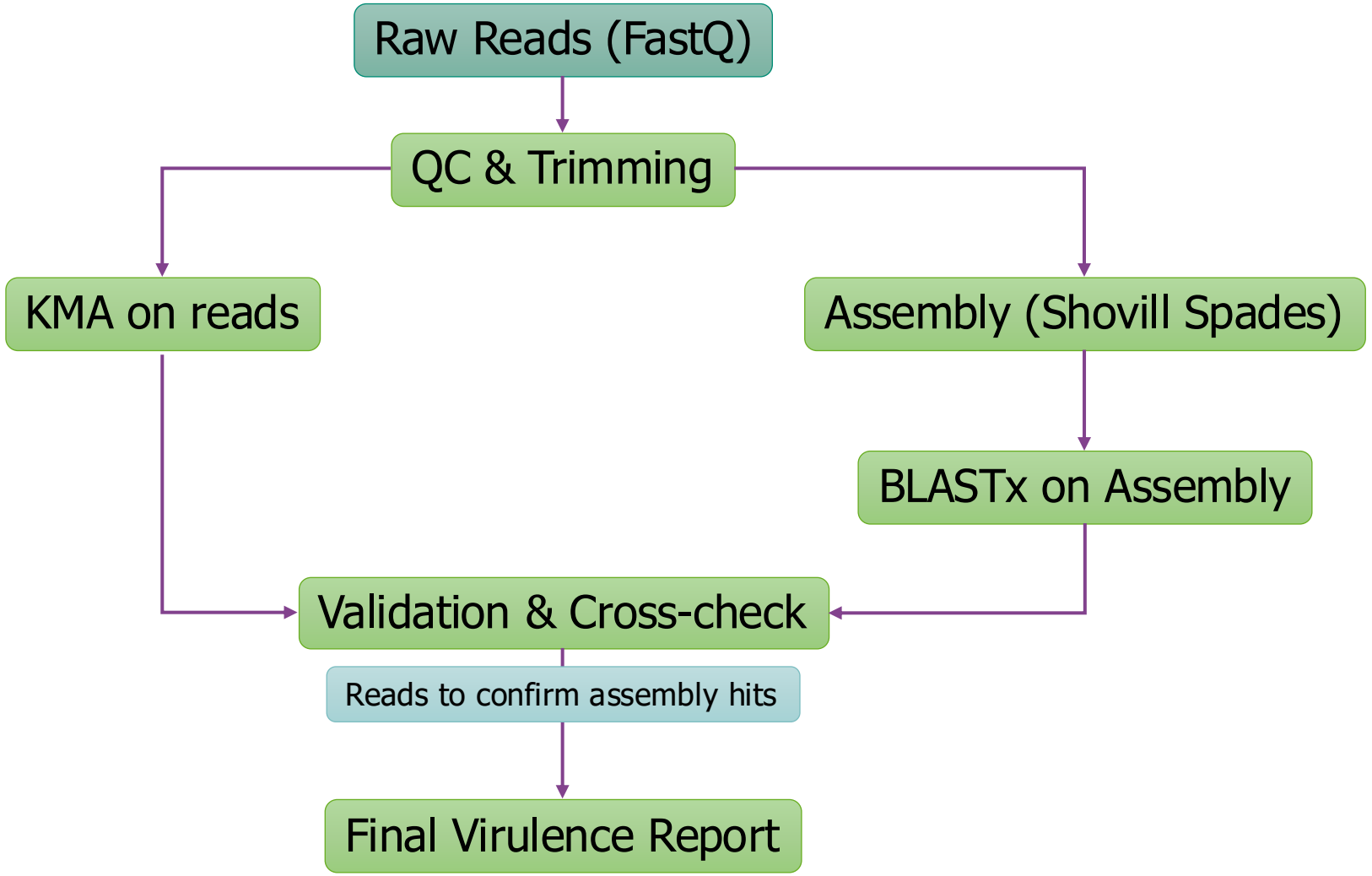
Step 5: Run KMA on Reads

```
kma \
    -ipe sample_R1_trimmed.fastq.gz sample_R2_trimmed.fastq.gz \
    -t_db /data/databases/Virulence/vfdb/vfdb_setB_nt_kma \
    -o    /data/out/SAMPLE/vfdb_kma_reads \
    -t    8 \           # Threads
    -1t1 \              # One template, one match (best hit per read)
    -mem_mode \         # Use memory-efficient mode for large DBs
    -ID 90 \            # Minimum 90% nucleotide identity
    -md 5               # Minimum depth of 5x coverage
```

Step 6: Run BLASTx on Assembly

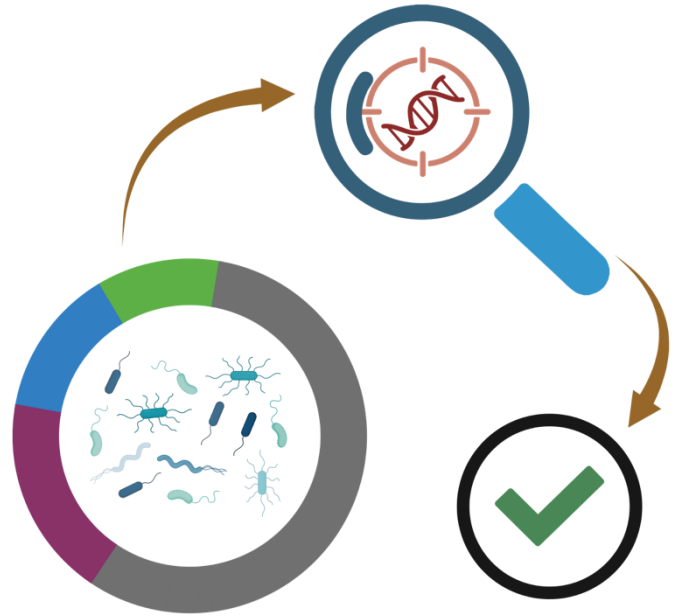
```
blastx \
    -query  /data/out/SAMPLE/Assembly/shovill_spades/SAMPLE/contigs.fa \
    -db     /data/databases/Virulence/vfdb/vfdb_setB_pro \
    -outfmt "6 qseqid sseqid pident length evalue bitscore qstart qend sstart send stitle" \
    -max_target_seqs 1 \    # Keep only best protein hit per contig region
    -evalue 1e-10 \         # Strict e-value cutoff
    -num_threads 8 \
    -out    /data/out/SAMPLE/Virulence/SAMPLE/vfdb_blastx.tsv
```

# How Can We Use Database with Command Line?



# Advantages on Combining Read-Assembly?

- Higher sensitivity and resilience to poor assemblies
- Cross-validation and quality control of calls
- Better linkage of VFs to genomic context
- More complete virulence profiling for surveillance



# Acknowledgements

**The creation of this training material was commissioned by ECDC to Technical University of Denmark (DTU) with the direct involvement of João Cardoso (Research Assistant at DTU, Msc – joacar@food.dtu.dk)**