

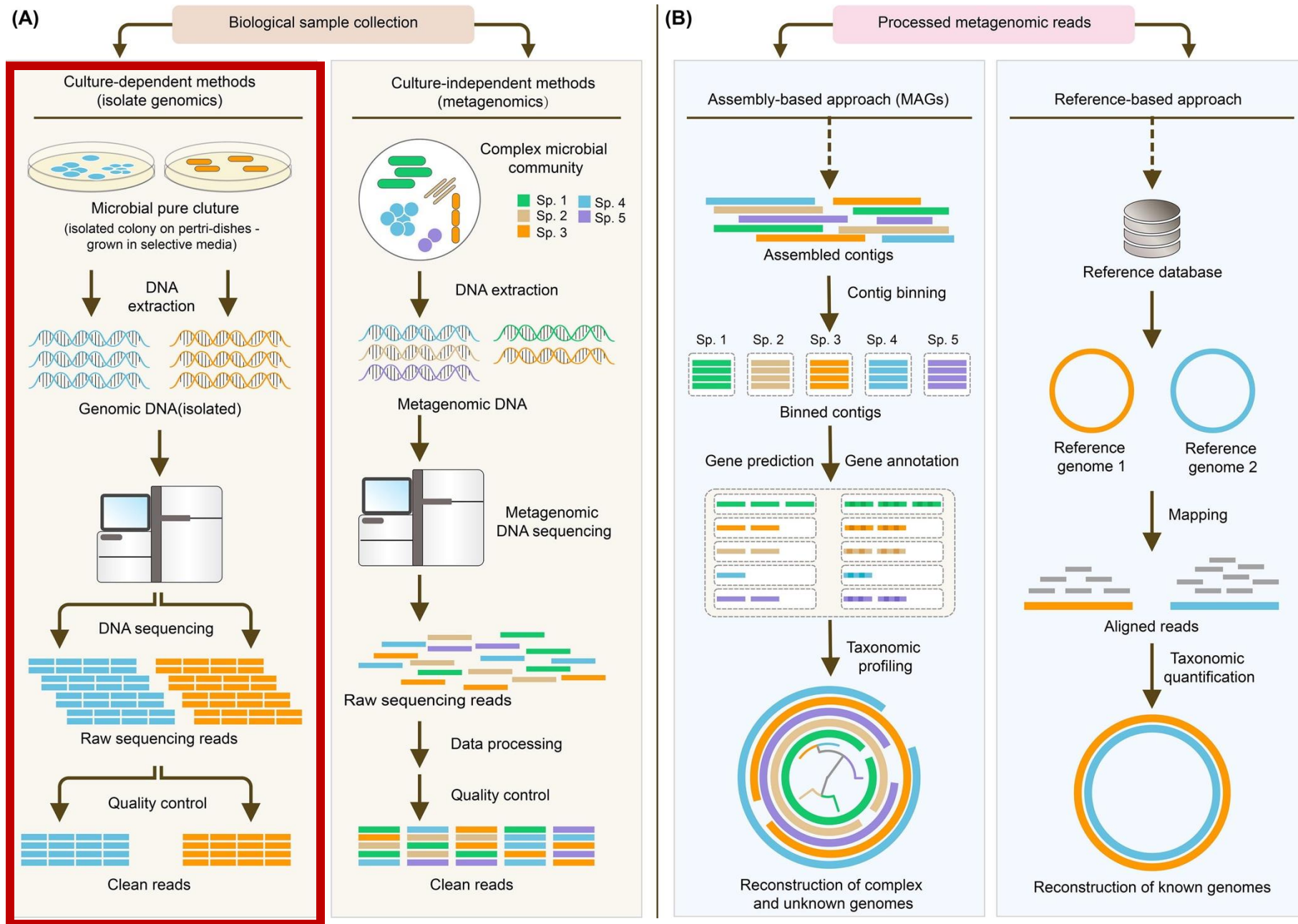
From isolates to assemblies for Illumina Short Reads

A pipeline of the Vonaesch Lab for the curnagl
cluster

Simon Yersin

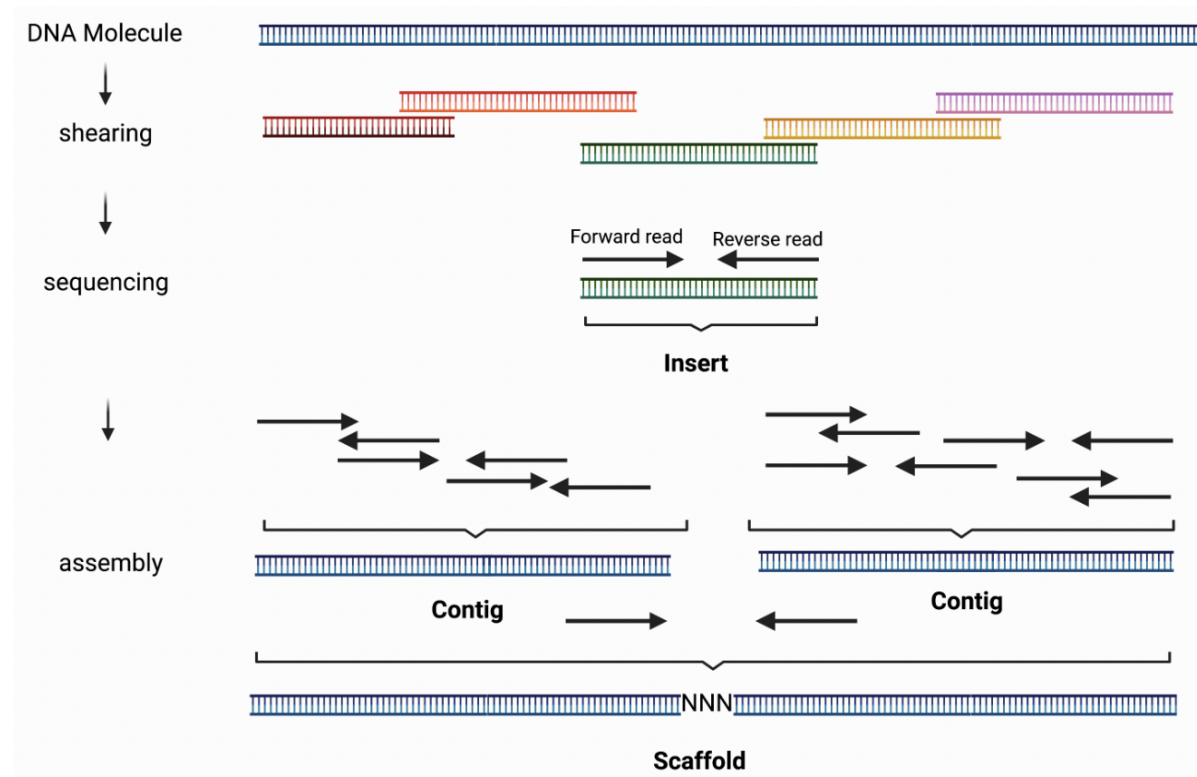
Introduction

- This bioinformatic pipeline slides attempts to describe the steps to go from Illumina short reads of bacterial isolates to assembled genomes.



Genome assembly

While powerful, short read sequencing technologies only allow assembly of draft genomes (i.e. assembly consisting of multiple scaffolds). As illustrated below, during whole genome shotgun sequencing, DNA is randomly sheared into inserts of known size distribution and sequenced. If paired-end sequencing is used, two DNA sequences (reads) are generated - one from each end of a DNA fragment). The assemblers look for overlaps between sequencing reads to stitch them together into contigs. The contigs can then sometimes be linked together into longer scaffolds.

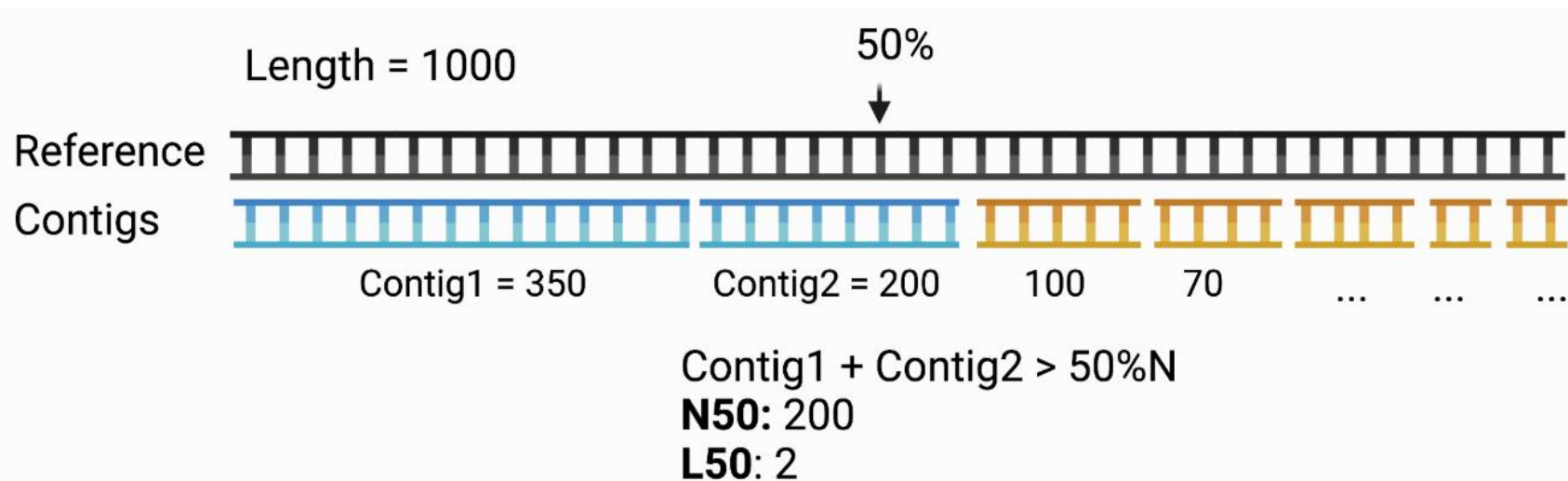


Metrics for high quality genome

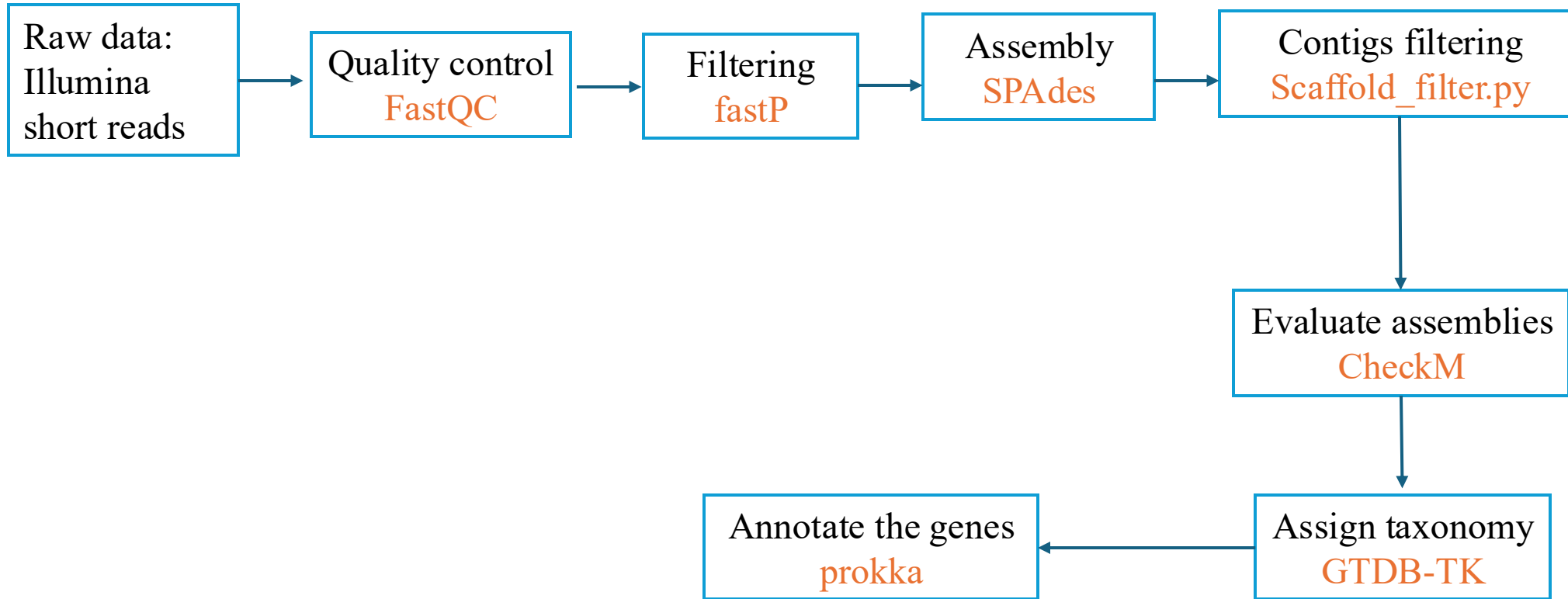
- Contiguity: we want fewer contigs and long contigs for a continuous assembly
- Completeness: total size
- Correctness: Proportion of the assembly that is free of mistakes

N50: size of the shortest contig covering 50% of the total assembly → we want higher

L50: number of contig making up 50% of the cumulated contig length → we want small



General pipeline



STEP 0: PREPARATION

➤ Prepare your working directory with sub-directories:

 Shortreads_files

 Raw_data

 Scripts

 Outputs

➤ Then copy your raw data from the nas to Raw_data folder:

 `cp /nas/FAC/FBM/DMF/pvonaesc/default/D2c/Simon\ Yersin/<raw data directory> ~/S_salivarius_shortreads/Raw_data`

➤ Wait for the copy to finish and verify with **ls** that all your files are copied

STEP 1: Quality Control

- Perform the quality assessment of your reads:

 01_1_fastqc.sh

- Adapt and use read_count.sh to save a file with the number of reads before filtering

 01_2_read_count.sh

STEP 2: FILTERING

- Filter your reads using *fastP*:

 02_1_fastp.sh

- Adapt and use `read_count.sh` to save a file with the number of reads after filtering

 02_2_read_count.sh

STEP 3: ASSEMBLY

- Assemble your genome using *SPAdes*:

 `03_spades.sh`

- Adapt variables and array
- Follow the progress of the job with *Squeue*, *sacct* or by opening the log file of the assembly

 (*OPTION*) Download assembly graph (assembly_graph.gfa) and open it on *Bandage*

- File > load graph
- Draw graph
- Save assembly graph

STEP 4: Filter contigs

- Filter the contigs and scaffolds using the python script `scaffold_filter.py` and the following script:



 `04_scaffold_filter.sh`

- We are producing fasta files filtered at 200 and 1000 minimum length for both the contigs and the scaffolds.
- We are also producing statistic files to evaluate the assemblies.


See methods in microbiomics for the explanations and python script:

https://methods-in-microbiomics.readthedocs.io/en/latest/assembly/genome_assembly.html)

STEP 5: Evaluate the assemblies

- Copy final assemblies into new folder using:
 `05_1_copy_assembly.sh`
- Evaluate the quality of your assemblies using *CheckM* :
 `05_2_checkM.sh`
- Download report: `CheckM_QC_stats.tsv`

STEP 6: GTDB-TK

- GTDB-TK is a software toolkit for assigning objective taxonomic classifications to bacterial and archaeal genomes based on the Genome Database Taxonomy.
- Run GTDB-TK to obtain the taxonomic classification of your genomes, using:
 `06_gtdb_tk.sh`
- Download the output file: `gtdbtk.bac120.summary.tsv`

Downstream steps and analysis

- From here we have assembled, filtered, QC controlled, and taxonomy assigned genomes from Illumina short reads
- The next steps depend on the research question, here are some options:
 - Genes annotation and comparative genomics (Anv'io, discuss with Julian)
 - Strain diversity analysis using fastANI or inStrain
 - Phylogeny

FINAL NOTES

- The standard output files .err and .out can be erased once runs are completed without failures or errors
- Download all scripts files or save them on the NAS for reproducibility
- Download output files needed for downstream analysis
- Clean conda with: `conda clean --tarballs --packages`

Good job!