

## Bacterial hybrid assembly

de novo assembly; zuerst wird ein short-read assembly gemacht mit SPADES und dann scaffolding mit long reads

- Nanopore und Illumina fastq files in den Projektordner unter ncct-projects kopieren (fastq-ilmn bzw fastq-ont)
- Die ont-files muss man noch mergen mit dem cat-Befehl: in den entsprechenden Ordner gehen (z.B. `cd fastq-ont/barcode15`) und dann eingeben: `cat PAE*.fastq > barcode15.fastq`
- Die gemergten Files in den fastq-ont Ordner raus kopieren

### Mit nfcore-pipeline

Funktioniert manchmal nicht; es kann auch Fehler geben, die nicht gemeldet werden, daher immer im log-file nachschauen ob es ohne Fehler durchgelaufen ist!

#### 1) Sample Sheet erstellen

- unter Windows neuen Ordner erstellen mit den fastq-files und dem samplesheet.tsv (tab-separated) - erklärt auch auf der Seite von nf-core/bacass unter Parameters
- Sample Sheet in Excel erstellen. Struktur:  
ID      R1      R2      LongFastQ      Fast5      GenomeSize

Diese Bezeichnungen kommen in Zeile 1

Die Filenamen kann man für größere Probenzahl erstellen, indem man in einer weiteren Spalte fastq-ilmn/ eingibt (den Namen des Ordners mit den fastq-files) und dann mit VERKETTEN() zusammenfügt

Die fastq-file IDs kann man sich über `ls fastq-file-folder > test` als .txt-Datei ausgeben lassen und dann in Excel kopieren

GenomeSize abschätzen

- Alles aus Excel rauskopieren und in einem neuen Excel-File nur als Werte einfügen
- Abspeichern als .txt
- Öffnen mit Notepad+
- Trennzeichen wenn nötig ersetzen durch `\t` (über suchen - ersetzen; unter "Erweitert" `\t` auswählen)
- Unten rechts das Format "CR LF" ändern in Unix (LF); außerdem muss es UTF-8 sein
- Abspeichern als samplesheet.tsv

#### 2) Nextflow-Pipeline

- `cd` zum Projektordner
- `nextflow pull nf-core/bacass`
- `nextflow run nf-core/bacass --input samplesheet.tsv --kraken2db /home/sysgen/db/kraken/k2_standard_8gb_20200919 --annotation_tool dfast --assembly_type hybrid -profile docker`
- nach Beendigung ins unicycler.log pro Probe schauen; bei Error die manuelle Variante machen

### 3) QC mit multiqc und Quast

Siehe beim manuellen Assembly

Varianten: Illumina oder Nanopore only Assembly - einfach beim nicht vorhandenen im sample sheet NA angeben und bei assembly\_type short oder long

#### Unicycler manuell

github Unicycler quick usage für Infos: <https://github.com/rwick/Unicycler#quick-usage>  
unicycler -h für Hilfe und Parameter

#### 1) Sample Sheet erstellen

- unter Windows neuen Ordner erstellen mit den fastq-files und dem samplesheet.csv (comma separated)
- Sample Sheet in Excel erstellen. Struktur:  
ID,R1,R2,LongFastQ

Diese Bezeichnungen kommen **nicht** in Zeile 1 (keine Header)!!

Die Filenamen kann man für größere Probenzahl erstellen, indem man in einer weiteren Spalte fastq-ilmn/ eingibt (den Namen des Ordners mit den fastq-files) und dann mit VERKETTEN() zusammenfügt

Die fastq-file IDs kann man sich über ls fastq-file-folder > test als .txt-Datei ausgeben lassen und dann in Excel kopieren

- Alles aus Excel rauskopieren und in einem neuen Excel-File nur als Werte einfügen
- Abspeichern als .txt
- Öffnen mit Notepad+
- Trennzeichen (\t) wenn nötig ersetzen durch , (Über Suchen - Ersetzen)
- Unten rechts das Format "CR LF" ändern in Unix (LF); außerdem muss es UTF-8 sein (Unter "Kodierung")
- Abspeichern als samplesheet.csv

#### 2) Unicycler manuell

- cd in den Projektordner
- `conda activate unicycler`
- `while IFS="," read sample read1 read2 longread; do unicycler -1 $read1 -2 $read2 -l $longread -o $sample; done < samplesheet.csv`

Achtung: Bei vielen Proben die Threads hochsetzen mit `-t 80` (hinter `-o $sample`; max Threads sind 112)

- nach Beendigung: `conda deactivate`
- Ergebnisordner in neuen Ordner schieben (results-unicycler)

#### 3) QC mit Quast

- `conda activate quast`
- `quast results-unicycler/*/assembly.fasta` ("unicycler-results" ist der name des Überordners)
- nach Beendigung: `conda deactivate`

#### 4) MultiQC (ohne conda)

- `multiqc quast_results/latest`

## PlasmIDent Pipeline

Nach Unicycler, Quast und MultiQC

Anleitung unter <https://github.com/imgag/plasmIDent>

### 1) Sample Sheet erstellen

- Sample Sheet in Excel erstellen. Struktur:

sample_id	assembly_fasta	longread_fq
-----------	----------------	-------------

Bsp: myid1	path/to/assembly1.fasta	path/to/reads1.fastq
------------	-------------------------	----------------------

Die Bezeichnungen kommen **nicht** in Zeile 1 (keine Header)!!

Die Filenamen kann man für größere Probenzahl erstellen, indem man in einer weiteren Spalte unicycler-result/ eingibt (den Namen des Ordners mit den assembly-files) plus eine weitere Spalte mit /assembly.fasta und dann mit VERKETTEN() zusammenfügt

Die IDs kann man sich über ls fastq-file-folder > test als .txt-Datei ausgeben lassen und dann in Excel kopieren

- Alles aus Excel rauskopieren und in einem neuen Excel-File nur als Werte einfügen
- Abspeichern als .txt
- Öffnen mit Notepad+
- Trennzeichen wenn nötig ersetzen durch \t (über suchen - ersetzen; unter "Erweitert" \t auswählen)
- Unten rechts das Format "CR LF" ändern in Unix (LF); außerdem muss es UTF-8 sein
- Abspeichern als samplesheet.tsv

### 2) Pipeline

- cd in den Projektordner
- nextflow pull imgag/plasmIDent
- nextflow run imgag/plasmIDent --input samplesheet.tsv --outDir result-plasmident -profile docker

Als Ergebnisse gibt es einen Ergebnisordner pro Stamm, darunter bei "plots" die Bilder (fängt bei \_2 an, da \_1 immer das Chromosom ist); bei "plasmids" die fasta files der Plasmide, zudem Antibiotika-Resistenz-Gene