ontpipeline_document_de Documentation

Yan Zhou

13.06.2019

Inhaltsverzeichnis

KAPITEL 1

Installation

1.1 Installation

1.1.1 Anaconda Installation

Installing on Linux https://docs.anaconda.com/anaconda/install/linux/

Bemerkung:

- Anaconda is installed in `/opt` directory .

1.1.2 JDK8 Installation[9]

1. Download source pakage from Oracle. https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
2. Extract JDK8 files to the target folder.

```
sudo mkdir /usr/lib/jvm
sudo tar -zxvf jdk-8u211-linux-x64.tar.gz -C /usr/lib/jvm
```

3. Set environment variables for JDK8.

```
sudo nano ~/.bashrc
#set oracle jdk environment
export JAVA_HOME=/usr/lib/jvm/jdk-1.8.0_211
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

(Fortsetzung auf der nächsten Seite)

[9] Ubuntu JDK 7 / JDK8 https://www.cnblogs.com/a2211009/p/4265225.html

(Fortsetzung der vorherigen Seite)

```
#make changes take effect immediately
source ~/.bashrc
```

4. Set JDK8 to jdk-1.8.0_211.

```
sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-1.8.0_211/bin/
↪java 300
sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk-1.8.0_211/
↪bin/javac 300
sudo update-alternatives --install /usr/bin/jar jar /usr/lib/jvm/jdk-1.8.0_211/bin/
↪jar 300
sudo update-alternatives --install /usr/bin/javah javah /usr/lib/jvm/jdk-1.8.0_211/
↪bin/javah 300
sudo update-alternatives --install /usr/bin/javap javap /usr/lib/jvm/jdk-1.8.0_211/
↪bin/javap 300
#set path to jdk-1.8.0_211
sudo update-alternatives --config java
```

5. Test

```
java -version
# The following messages should be showed if it works.
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
```

1.1.3 Guppy Installation

Guppy is a data processing toolkit that contains the Oxford Nanopore Technologies‘ basecalling algorithms, and several bioinformatic post-processing features.[1]

```
wget https://mirror.oxfordnanoportal.com/software/analysis/ont-guppy-cpu_3.0.3_
↪linux64.tar.gz
tar -xf ont-guppy-cpu_3.0.3_linux64.tar.gz
sudo mv ont-guppy-cpu_3.0.3_linux64 /opt/ont-guppy-cpu_3.0.3
```

1.1.4 Porechop Installation

Porechop is a tool for finding and removing adapters from Oxford Nanopore reads.[2]

```
/opt/anaconda3/bin/conda create -n porechop
source /opt/anaconda3/bin/activate porechop
conda install -c bioconda porechop
conda deactivate
```

1.1.5 NanoStat Installation

NanoStat calculates various statistics from a long read sequencing dataset in fastq, bam or albacore sequencing summary format.[3]

[1] Guppy v3.0.3 Release https://community.nanoporetech.com/posts/guppy-3-0-release
[2] Porechop https://github.com/rrwick/Porechop
[3] NanoStat https://github.com/wdecoster/nanostat

```
/opt/anaconda3/bin/conda create -n nanostat
source /opt/anaconda3/bin/activate nanostat
conda install -c bioconda nanostat
conda deactivate
```

1.1.6 NanoFilt Installation

NanoFilt filters and trims long read sequencing data.[4]

```
/opt/anaconda3/bin/conda create -n nanofilt
source /opt/anaconda3/bin/activate nanofilt
conda install -c bioconda nanofilt
conda deactivate
```

1.1.7 Unicycler Installation

Unicycler is an assembly pipeline for bacterial genomes.[5]

```
/opt/anaconda3/bin/conda create -n unicycler
source /opt/anaconda3/bin/activate unicylcer
conda install -c bioconda unicycler
conda install -c bioconda bcftools # for .vcf file
conda deactivate
```

Warnung:

- Change the memory setting in Pilon is necessary or it could be failed to start[10] .

1.1.8 BUSCO Installation

BUSCO v3 provides quantitative measures for the assessment of genome assembly, gene set, and transcriptome completeness, based on evolutionarily-informed expectations of gene content from near-universal single-copy orthologs selected from OrthoDB v9.[6]

```
/opt/anaconda3/bin/conda create -n busco
source /opt/anaconda3/bin/activate busco
conda install -c bioconda busco
conda deactivate
```

1.1.9 BWA Installation

BWA is a software package for mapping low-divergent sequences against a large reference genome.[7]

[4] NanoFilt https://github.com/wdecoster/nanofilt
[5] Unicycler https://github.com/rrwick/Unicycler
[10] Pilon step runs out of error https://github.com/rrwick/Unicycler/issues/147
[6] BUSCO v3 https://busco.ezlab.org
[7] BWA https://github.com/lh3/bwa

```
/opt/anaconda3/bin/conda create -n bwa
source /opt/anaconda3/bin/activate bwa
conda install -c bioconda bwa
conda deactivate
```

1.1.10 Seqtk Installation

Seqtk is a fast and lightweight tool for processing sequences in the FASTA or FASTQ format.[8]

```
/opt/anaconda3/bin/conda create -n seqtk
source /opt/anaconda3/bin/activate seqtk
conda install -c bioconda seqtk
conda deactivate
```

[8] Seqtk https://github.com/lh3/seqtk

KAPITEL 2

Die Struktur der Eingabedaten

2.1 Start von Base Calling

Start die Pipeline von Base Calling.

```
ONT_Reads_Directory/
├── HPz800_20180821_FAJ18422_MN17776_sequencing_run_VIM4_Janina_26844_read_11_ch_171_
↪strand.fast5
├── HPz800_20180821_FAJ18422_MN17776_sequencing_run_VIM4_Janina_26844_read_11_ch_203_
↪strand.fast5
├── HPz800_20180821_FAJ18422_MN17776_sequencing_run_VIM4_Janina_26844_read_15_ch_344_
↪strand.fast5
├── HPz800_20180821_FAJ18422_MN17776_sequencing_run_VIM4_Janina_26844_read_17_ch_249_
↪strand.fast5
├── HPz800_20180821_FAJ18422_MN17776_sequencing_run_VIM4_Janina_26844_read_19_ch_397_
↪strand.fast5
└── ......

Illumina_Reads_Directory/
├── Präfix01_HQ_1.fastq.gz
├── Präfix01_HQ_2.fastq.gz
├── Präfix02_HQ_1.fastq.gz
├── Präfix02_HQ_2.fastq.gz
├── Präfix03_HQ_1.fastq.gz
├── Präfix03_HQ_2.fastq.gz
└── ......
```

Bemerkung:

- Die Benennungsregeln für jedes Illumina-Reads Paar: Präfix_HQ_1.fastq.gz Präfix_HQ_2.fastq.gz
- „Präfix“ ist der Probenname und das ist identisch für jedes Paar.
- „*“ bedeuted beliebig lange Zeichen.

Warnung:

- Unterstrich(‚_‘) ist im Präfix nicht erlaubt.

2.2 Start von Demultiplexing

Start die Pipeline von Demultiplexing.

```
ONT_Reads_Directory/
├── fastq_runid_50a6171cadcfb6b5cb2dae4e75a0ccc05b71e3d8_0.fastq
├── fastq_runid_50a6171cadcfb6b5cb2dae4e75a0ccc05b71e3d8_1.fastq
├── fastq_runid_50a6171cadcfb6b5cb2dae4e75a0ccc05b71e3d8_2.fastq
├── fastq_runid_50a6171cadcfb6b5cb2dae4e75a0ccc05b71e3d8_3.fastq
├── fastq_runid_50a6171cadcfb6b5cb2dae4e75a0ccc05b71e3d8_4.fastq
└── ......

Illumina_Reads_Directory/
├── Präfix01_HQ_1.fastq.gz
├── Präfix01_HQ_2.fastq.gz
├── Präfix02_HQ_1.fastq.gz
├── Präfix02_HQ_2.fastq.gz
├── Präfix03_HQ_1.fastq.gz
├── Präfix03_HQ_2.fastq.gz
└── ......
```

2.3 Start von Reads Filter

Start die Pipeline von Reads Filter.

```
ONT_Reads_Directory/
├── Präfix01.fastq
├── Präfix02.fastq
├── Präfix03.fastq
├── Präfix04.fastq
├── Präfix05.fastq
└── ......

Illumina_Reads_Directory/
├── Präfix01_HQ_1.fastq.gz
├── Präfix01_HQ_2.fastq.gz
├── Präfix02_HQ_1.fastq.gz
├── Präfix02_HQ_2.fastq.gz
├── Präfix03_HQ_1.fastq.gz
├── Präfix03_HQ_2.fastq.gz
└── ......
```

2.4 Start von Assembly

Start die Pipeline von Assembly.

```
ONT_Reads_Directory/
├── Präfix01.fastq
├── Präfix02.fastq
├── Präfix03.fastq
├── Präfix04.fastq
├── Präfix05.fastq
└── ......

Illumina_Reads_Directory/
├── Präfix01_HQ_1.fastq.gz
├── Präfix01_HQ_2.fastq.gz
├── Präfix02_HQ_1.fastq.gz
├── Präfix02_HQ_2.fastq.gz
├── Präfix03_HQ_1.fastq.gz
├── Präfix03_HQ_2.fastq.gz
└── ......
```

2.5 Start von Polishing

Start die Pipeline von Polishing.

```
ONT_Reads_Directory/
├── Präfix01.fasta
├── Präfix02.fasta
├── Präfix03.fasta
├── Präfix04.fasta
├── Präfix05.fasta
└── ......

Illumina_Reads_Directory/
├── Präfix01_HQ_1.fastq.gz
├── Präfix01_HQ_2.fastq.gz
├── Präfix02_HQ_1.fastq.gz
├── Präfix02_HQ_2.fastq.gz
├── Präfix03_HQ_1.fastq.gz
├── Präfix03_HQ_2.fastq.gz
└── ......
```

2.6 Musterblatt(Sample Sheet)

Tab. 1: Musterblatt

Probenname	Barcode
Probenname1	barcode01
Probenname2	barcode02
Probenname3	barcode03
Probenname4	barcode04
Probenname5	barcode05

Bemerkung:

- Das Dateiformat des Musterblattes ist entweder CSV (Trennzeichen ist das Komma) oder TSV (Trennzeichen ist die Tabulatortaste).
- Die Benennungsregeln für Barcode: barcodeXX („barcode“ kann beliebig Zeichen sein, aber „XX“ muss zweistellig sein,z.B. 01,02,03,. . . ,10,11,12,. . .)

KAPITEL 3

Die Struktur der Ausgabedaten

```
Output_Directory/
 ├── Analysis_{Zeitstempel}/
 |     ├── Basecalled/
 |     ├── Barcodes/
 |     |     ├── barcode01/
 |     |     ├── barcode02/
 |     |     ├── barcode03/
 |     |     ├── unclassified/
 |     |     ├── Präfix01.fastq
 |     |     ├── Präfix02.fastq
 |     |     └── Präfix03.fastq
 |     ├── AdaperTrimmedFiles/
 |     |     ├── Präfix01_trimmed.fastq
 |     |     ├── Präfix02_trimmed.fastq
 |     |     └── Präfix03_trimmed.fastq
 |     ├── FiltedFiles/
 |     |     ├── Präfix01_filted.fastq
 |     |     ├── Präfix02_filted.fastq
 |     |     └── Präfix03_filted.fastq
 |     ├── StatFiles/
 |     |     ├── Präfix01_trimmed_stat.txt
 |     |     ├── Präfix02_trimmed_stat.txt
 |     |     ├── Präfix03_trimmed_stat.txt
 |     |     ├── Präfix01_filted_stat.txt
 |     |     ├── Präfix02_filted_stat.txt
 |     |     └── Präfix03_filted_stat.txt
 |     ├── Präfix01_Assembly/
 |     |     ├── ...
 |     |     └── assembly.fasta
 |     ├── Präfix02_Assembly/
 |     |     ├── ...
 |     |     └── assembly.fasta
 |     ├── Präfix03_Assembly/
 |     |     ├── ...
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

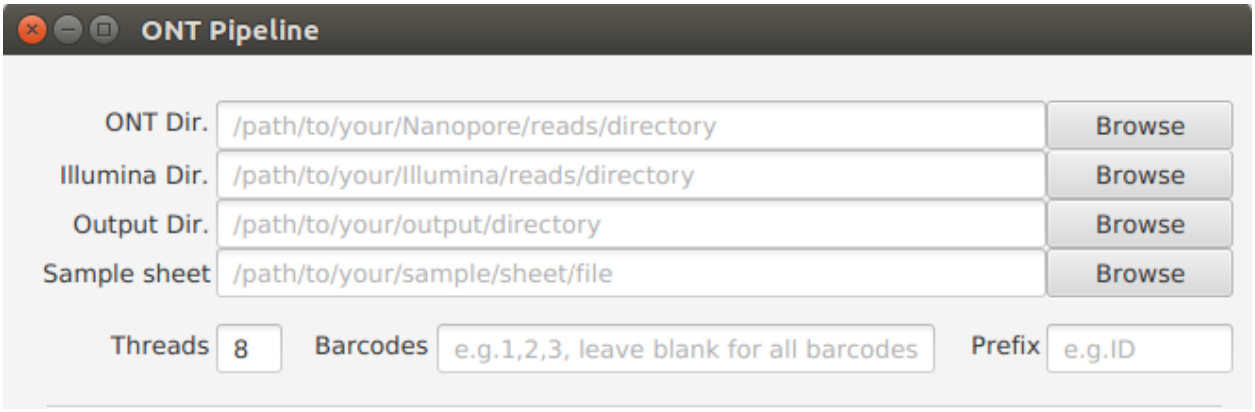
```
|     |     └── assembly.fasta
|     ├── Präfix01_Polishing/
|     |     ├── run_Präfix01_busco/
|     |     |     ├── ...
|     |     |     └── full_table_Präfix01_busco.tsv
|     |     ├── ...
|     |     └── pilon_1.fasta
|     ├── Präfix02_Polishing/
|     |     ├── run_Präfix02_busco/
|     |     |     ├── ...
|     |     |     └── full_table_Präfix02_busco.tsv
|     |     ├── ...
|     |     └── pilon_1.fasta
|     ├── Präfix03_Polishing/
|     |     ├── run_Präfix03_busco/
|     |     |     ├── ...
|     |     |     └── full_table_Präfix03_busco.tsv
|     |     ├── ...
|     |     └── pilon_1.fasta
|     └── Logs/
|           ├── guppy_basecaller.log
|           ├── guppy_barcoder.log
|           ├── Präfix01_trimmed.log
|           ├── Präfix02_trimmed.log
|           ├── Präfix03_trimmed.log
|           ├── Präfix01_filted.log
|           ├── Präfix02_filted.log
|           ├── Präfix03_filted.log
|           ├── Präfix01_assembly.log
|           ├── Präfix02_assembly.log
|           ├── Präfix03_assembly.log
|           ├── Präfix01_polishing_1.log
|           ├── Präfix02_polishing_1.log
|           ├── Präfix03_polishing_1.log
|           ├── Präfix01_busco.log
|           ├── Präfix02_busco.log
|           └── Präfix03_busco.log
├── pipelineWithLoop_{Zeitstempel}.pbs # Übermittelte PBS-Datei.
└── userlog_{Zeitstempel}.log # Vom Benutzer angegebene Parameter.

/home/{$USER}/
├── Ont_Pipeline.e* # Fehlermeldungen in dem Lauf.
└── Ont_Pipeline.o* # Nachrichten in dem Lauf.

/opt/ontpipeline/logs/
├── ...
└── {$USER}_error.log # Fehlermeldungen für das Java-Programm.
```

KAPITEL 4

Allgemein Einstelllung



4.1 ONT Verzeichnis(ONT Dir.) (Erfordlich)

Der Verzeichnispfad zu den Nanopore-Reads einzugeben.

Bemerkung:

- **Beispiel:** `/path/to/your/ONT/reads/directory`

4.2 ONT Verzeichnis(Illumina Dir.) (Optional/Erfordlich)

Der Verzeichnispfad zu den Illumina-Reads einzugeben.

Bemerkung:

- **Beispiel:** `/path/to/your/Illumina/reads/directory`
- Erfordlich wenn „hybrid-assembly“ oder/und „polishing“ ausgewählt wird/werden.

4.3 Ausgabeverzeichnis(Output Dir.) (Erfordlich)

Der Verzeichnispfad zu den Ausgaben einzugeben.

Bemerkung:

- **Beispiel:** `/path/to/your/output/directory`

4.4 Musterblatt(Sample sheet) (Optional)

Der Pfadname zum Musterblatt einzugeben.

Bemerkung:

- Das Dateiformat des Musterblattes muss CSV oder TSV sein.

Warnung:

- Unterstrich(‚_‘) ist im Probenname **nicht** erlaubt.

4.5 Präfix(Prefix) (Optional)

Ein Präfix für die Umbenennung der Nanopore-Reads nach „demultiplexing“ einzugeben.

Bemerkung:

- **Bespiel:** ID .
- Standardwert: barcode .

4.6 (Theads)Threads (Erfordlich)

Die benötige Anzahl der Threads/CPUs für den Pipeline-Lauf einzugeben.

Bemerkung:

- Standardwert: 8.

4.7 (Barcodes)Barcodes (Optional)

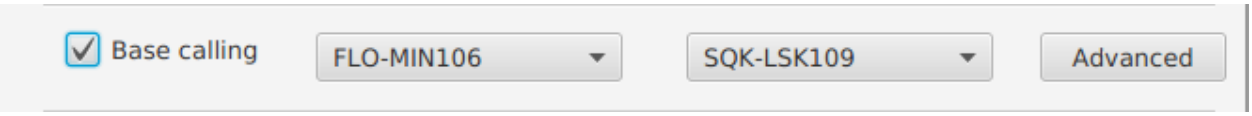
Welche Barcodes, die zum Pipeline-Lauf gebracht werden, einzugeben. Einfach die Barcode-Nummern, die mit dem Komma getrennt werden, einzugeben.

Bemerkung:

- **Beispiel:** 1,2,3,4
- Falls alle Barcodes zum Pipeline-Lauf gebracht werden, einfach dieses Feld leer lassen.

KAPITEL 5

Base Calling Einstellung

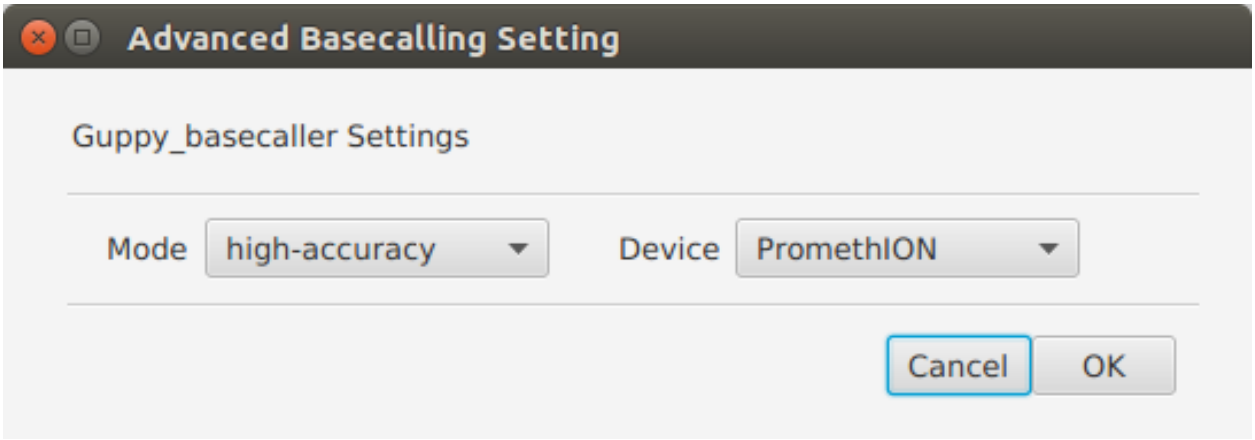


5.1 (Flowcell ID)Flowcell ID[2] (Erfordlich)

Eine Flowcell-ID aus der Liste auszuwählen.

5.2 (Kit Nummer)Kit Number[2] (Erfordlich)

Eine Kit-Nummer aus der Liste auszuwählen.



[2] How to configure Guppy parameters https://community.nanoporetech.com/protocols/Guppy-protocol-preRev/v/gpb_2003_v1_revg_14dec2018/how-to-configure-guppy-parameters

5.3 (Modus)Mode (Erfordlich)

Guppy „Base Calling“ Modus einzustellen.

Bemerkung:

- Standardwert: high-accuracy.

5.4 Gerät(Device) (Erfordlich)

Das Sequenzier-Geräte einzustellen.

Bemerkung:

- Standardwert: PromethION.

5.5 cpu_threads_per_caller[1] (Standardwert)

Bemerkung:

- Standardwert: 1.

5.6 records_per_fastq[2] (Standardwert)

Bemerkung:

- Standardwert: 0.
- Die Dateien werden per „Worker(CPU)“ und per „Run ID“ erstellt.

5.7 recursive[2] (Standardwert)

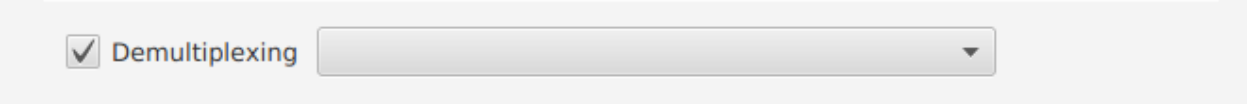
Bemerkung:

- Standardwert: die Eingabedateien werden rekursiv durchgesucht.

[1] Guppy v3.0.3 Release https://community.nanoporetech.com/posts/guppy-3-0-release

KAPITEL 6

Demultiplexing Einstellung



6.1 Barcode Kit(Barcode kit)[1] (Optional)

Einer Barcode-Kit oder mehrere Barcode-Kits aus der List auszuwählen, falls der/die verwendet wird/werden.

Bemerkung:

- Falls keiner Barcode-Kit verwendet wird, einfach dieses Feld leer lassen.
- Mehrfachauswahl ist möglich.

6.2 records_per_fastq[1] (Standardwert)

Bemerkung:

- Standardwert: 0.
- Die Dateien werden per „Worker(CPU)“ und per „Run ID“ erstellt.

[1] How to configure Guppy parameters https://community.nanoporetech.com/protocols/Guppy-protocol-preRev/v/gpb_2003_v1_revg_14dec2018/how-to-configure-guppy-parameters

6.3 recursive[1] (Standardwert)

Bemerkung:

- Standardwert: die Eingabedateien werden rekursiv durchgesucht.

6.4 trim_barcodes[2] (Default)

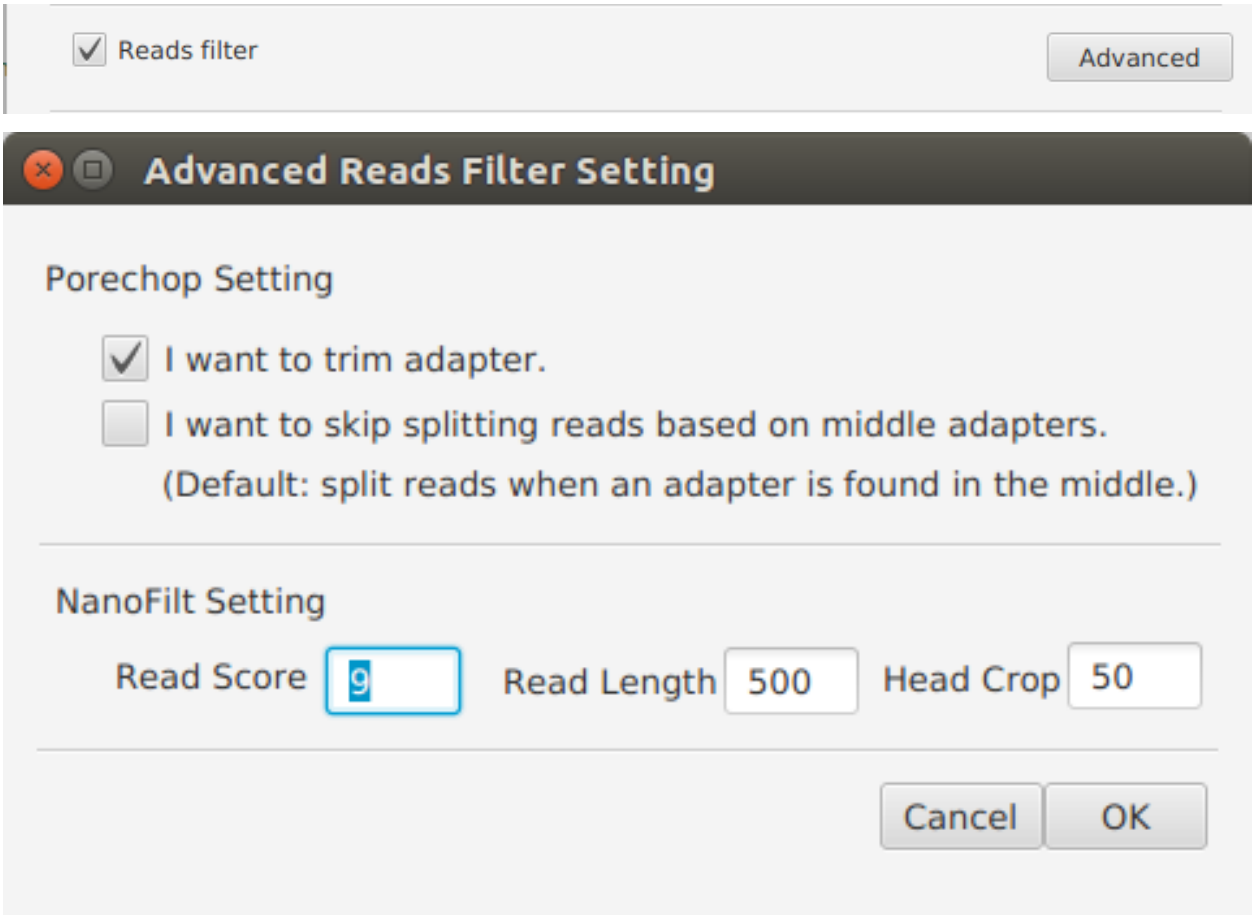
Bemerkung:

- Die Barcodes aus den Ausgabesequenzen in den FASTQ-Dateien werden geschnitten.

[2] Guppy update (v3.1.5) https://community.nanoporetech.com/posts/guppy-update-v3-1-5

KAPITEL 7

Reads Filter Einstellung



7.1 Porechop Einstellung[1] (Optional)

Die Optionen für Porechop einzustellen.

Bemerkung:

- „I want to trim adapter“ auszuwählen, wenn Sie Porechop verwenden möchten, um die Adapter der Sequenzen zu trimmen. Standardwert: ausgewählt.
- „I want to skip splitting reads based on middle adapters“ auszuwählen, wenn Sie keine Sequenz, die sich in der Mitte einen Adapter befindet, teilen möchten. Standardwert: nicht ausgewählt.

7.2 Read Score(Read Score)[2] (Erfordlich)

Einer durchschnittlichen Mindestwert für die Readqualität einzugeben, um die Reads zu filtern.

Bemerkung:

- Standardwert: 9.

7.3 Readlänge(Read Length) [2] (Erfordlich)

Eine minimale Readlänge einzugeben, um die Reads zu filtern.

Bemerkung:

- Standardwert: 500.

7.4 Kopf trimmen(Head Crop)[2] (Erfordlich)

Anzahl der Nukleotide, die ab dem Beginn des Reads geschnitten werden sollen, einzugeben.

Bemerkung:

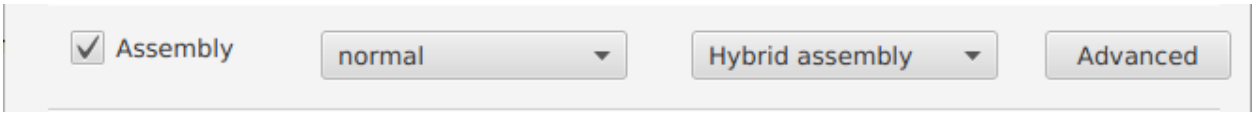
- Standardwert: 50.

---

[1] Porechop https://github.com/rrwick/Porechop
[2] NanoFilt https://github.com/wdecoster/nanofilt

KAPITEL 8

Assembly Einstellung



8.1 Modus(Mode)[1] (Erfordlich)

Ein Assember-Modus auszuwählen.

Bemerkung:

- Conservative: im Konservative-Modus wird das Assembly mit der geringsten Vollständigkeit und dem kleinsten Fehler erstellt.
- Normal: im Normal-Modus wird das Assembly mit mittlerer Vollständigkeit und mittlerem Fehler erstellt.
- Bold: im Grob-Modus wird das Assembly mit der höchsten Vollständigkeit und dem größten Fehler erstellt.
- Standardwert: Normal.

8.2 Methode(Method)[1] (Erfordlich)

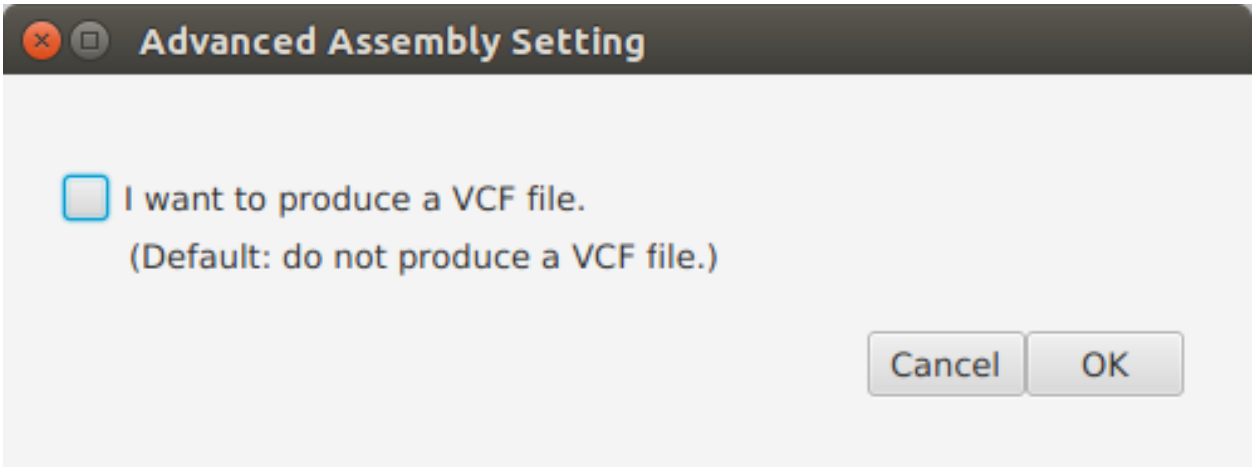
Eine Assember-Methode auszuwählen.

Bemerkung:

- Long-read-only assembly: mit der „Long-read-only assembly“-Methode werden nur die Nanopore-Reads zum Assembler gebracht.

[1] Unicycler https://github.com/rrwick/Unicycler

- Hybrid assembly: mir der „Hybrid assembly“-Methode werden Illumina-Reads und die Nanopore-Reads zum Assembler gebracht.
- Standardwert: Hybrid assembly.



8.3 VCF[1] (Optional)

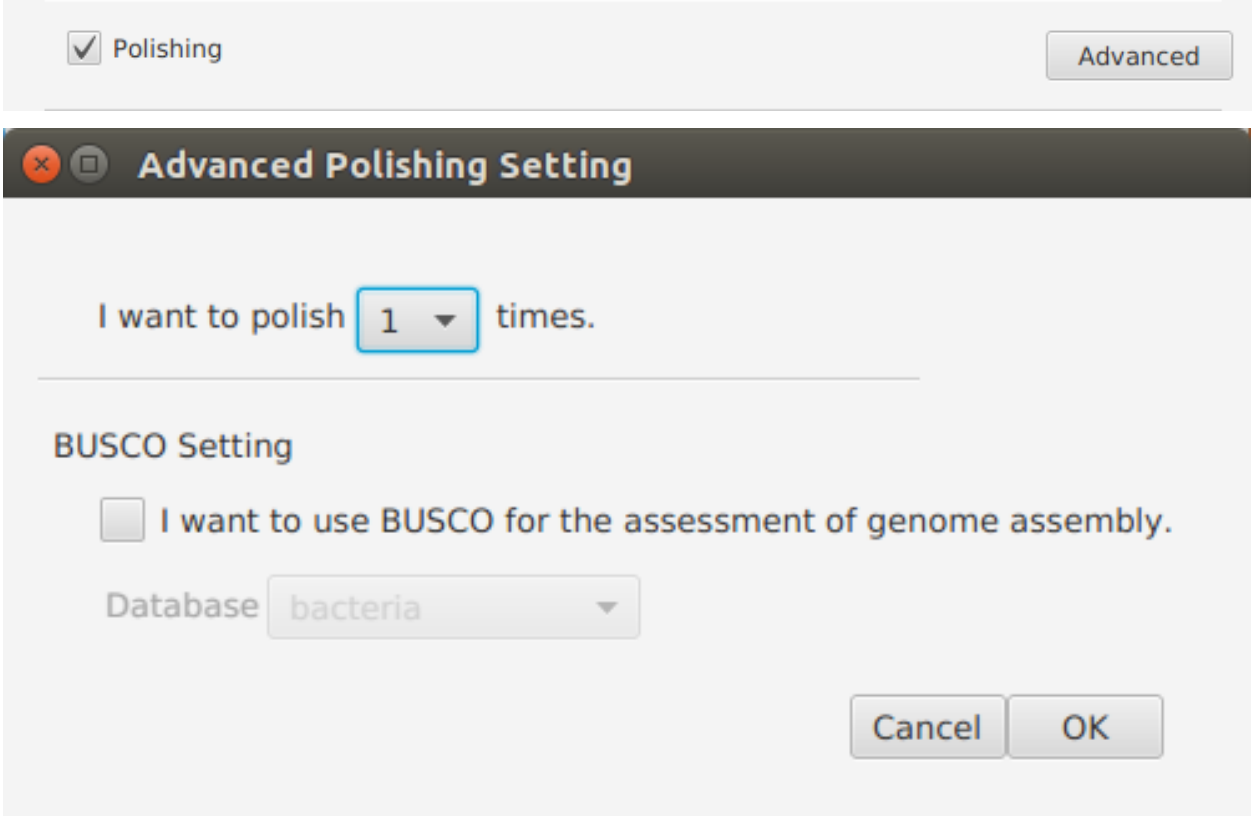
Eine VCF Datei wird erstellt, falls diese Option ausgewählt ist.

Bemerkung:

- Standardwert: nicht ausgewählt.

KAPITEL 9

Polishing Einstellung



9.1 Polishing Frequenz(Polishing times) (Erfordlich)

Die Anzahl der Frequenz für Polishing einzugeben.

Bemerkung:

- Standardwert: 1.

9.2 BUSCO Einstellung (Optional)

Die Optionen für BUSCO einzustellen.

Bemerkung:

- „I want to use BUSCO for the assessement of genome assembly“ auszuwählen wenn Sie BUSCO verwenden möchten. Standardwert: nicht ausgewählt.
- Einer Abstammungsdatensatz auszuwählen. Standardwert: Bacteria .

KAPITEL 10

FAQ

10.1 Welche Bioinformatik-Tools werden verwendet?

- Guppy https://community.nanoporetech.com
- Porechop https://github.com/rrwick/Porechop
- NanoStat https://github.com/wdecoster/nanostat
- NanoFilt https://github.com/wdecoster/nanofilt
- Unicycler https://github.com/rrwick/Unicycler
- BUSCO https://busco.ezlab.org
- Seqtk https://github.com/lh3/seqtk