sequ-into Documentation

Release latest

Apr 20, 2020

Contents

sequ-into is cool!

CHAPTER 1

sequ-into

1.1 A straightforward desktop app for third generation sequencing read analysis

Third generation sequencing techniques rapidly evolved as a common practice in molecular biology. Great advances have been made in terms of feasibility, cost, throughput, and read-length. However, sample contamination still poses a big issue: it complicates correct, high-quality downstream analysis of sequencing data and usage in medical applications.

To address these issues we developed a cross-platform desktop application: *sequ-into*. Reads originating from unwanted sources are detected and summarized by a comprehensive statistical overview, but can also be filtered and exported in standardized FASTQ-format to facilitate custom evaluation of experimental findings. Additionally, it might be unclear whether a sequencing experiment produced reads of the intended target. The filtering we implemented also allows for a positive selection of those reads who do.

sequ-into creates a straightforward user experience by fusing an intuitive graphical-user-interface with state-of-the-art long-read alignment software.

The app was implemented in the context of our *iGEM project Phactory*, where several DNA purification protocols were evaluated with *sequ-into* and thus allowed iterative engineering cycles leading to a so far unreached DNA purification of up to 96% (bases sequenced) in our probes. To learn more about Phactory, please follow this link: http://2018.igem.org/Team:Munich

1.2 Why is sequ-into an notable addition to your third generation sequencing routine?

"... our run times aren't fixed, unlike the other systems. Some people even have what they are looking for after a few minutes in real time, with success criteria not being based on total yield and an en-run analysis." **- Clive G. Brown, CTO of Oxford Nanopore**

This raises a simple question: How do you check if you have what we are looking for in a quick and easy manner?

The question could also be rephrased to: Is our sequencing run contaminated? In contrast to Illumina sequencing, in long read third generation sequencing (e.g PacBio or minION), there is always the possibility to abort a sequencing procedure, redo the library preparation and continue using the same chip. Especially when sequencing prokaryotic(-like) material, huge contaminations of the sample are possible. These could either be human DNA/RNA from the library prep, ribosomal RNA due to rRNA depletion not working, or even contamination from other organisms (host organism for phages, etc.).

Thus the earlier such contaminations are detected, the better the sequencing chip can be conserved for future use. Therfore we implemented *sequ-into*.

Especially since the library preparation and the sequencing are often in the hands of trained life scientists, while the down-stream analysis on the other hand is performed by trained computer scientists, there is often a gap in the workflow. *sequ-into* aims to close that gap as a convenient cross-platform tool, fusing an intuitive graphical-user-interface with state-of-the-art long-read alignment software.

1.3 What can sequ-into help you with and what not?

sequ-into is not an application for a thorough interpretive analysis of your sequencing data.

It can, however, be a very helpful addition to your sequencing lab routine. As it is easy to install and to use without much prior knowledge necessary, it is ideal for the very first assessment of your sequencing files.

In particular it allows you to quickly tell whether your sequenced reads represent your intended target or if your reads in fact stem from an unwanted source. Thus allowing for a fast reaction during long sequencing experiments and early alterations in your protocol in the laboratory prior to sequencing.

Later on, you might want to further investigate your sequencing data. Here, the extraction function of *sequ-into* offers the possibility to save reads of a wanted source or unwanted sources separately. Even the filtering by many possible contaminations at once is possible. *GraphMap*, the tool we employed for aligning reads to references, is highly sensitive and specialized for the utilization with third generation sequencing techniques.

CHAPTER 2

Installation

All pre-built binary releases can be found at GitHub .

However, the python scripts rely on additional software which must be made available.

2.1 Mac OS

To be able to install sequ-into on your Mac OS system, first make sure you have all support packages. **It is important to add them in this order.**

2.1.1 Support Package Installation:

1. **X-Code developer tools**

In oder to install python and required dependencies, you must prepare your Mac OS to have xcode-select tools installed (this is a package provided by Apple/Mac OS). Mac OS users can do so by running

```
xcode-select --install
```

in the terminal.

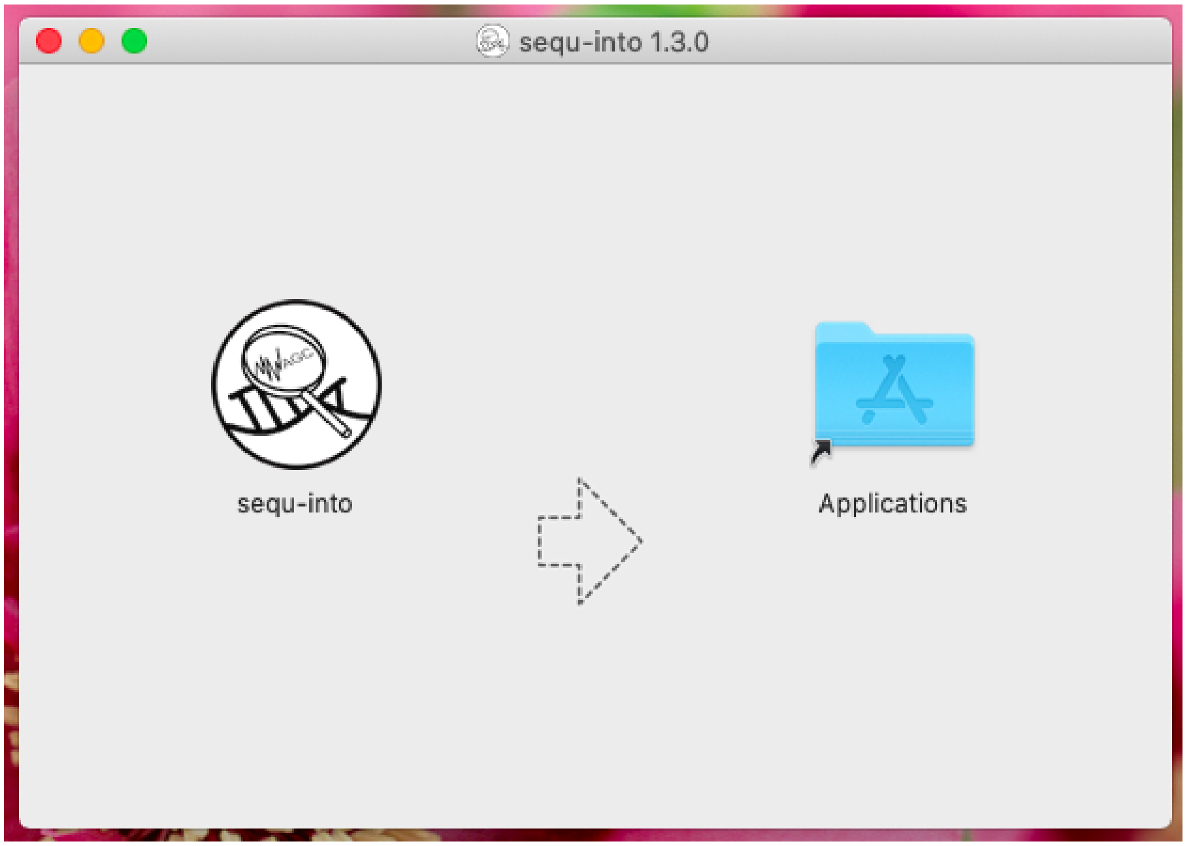
Follow the instructions displayed on your computer. Mac OS will download and install all required packages.

After having installed xcode-select, you can install sequ-into.

2.1.2 Install sequ-into

In order to install sequ-into, go to the sequ-into repository and download the Mac OS release.

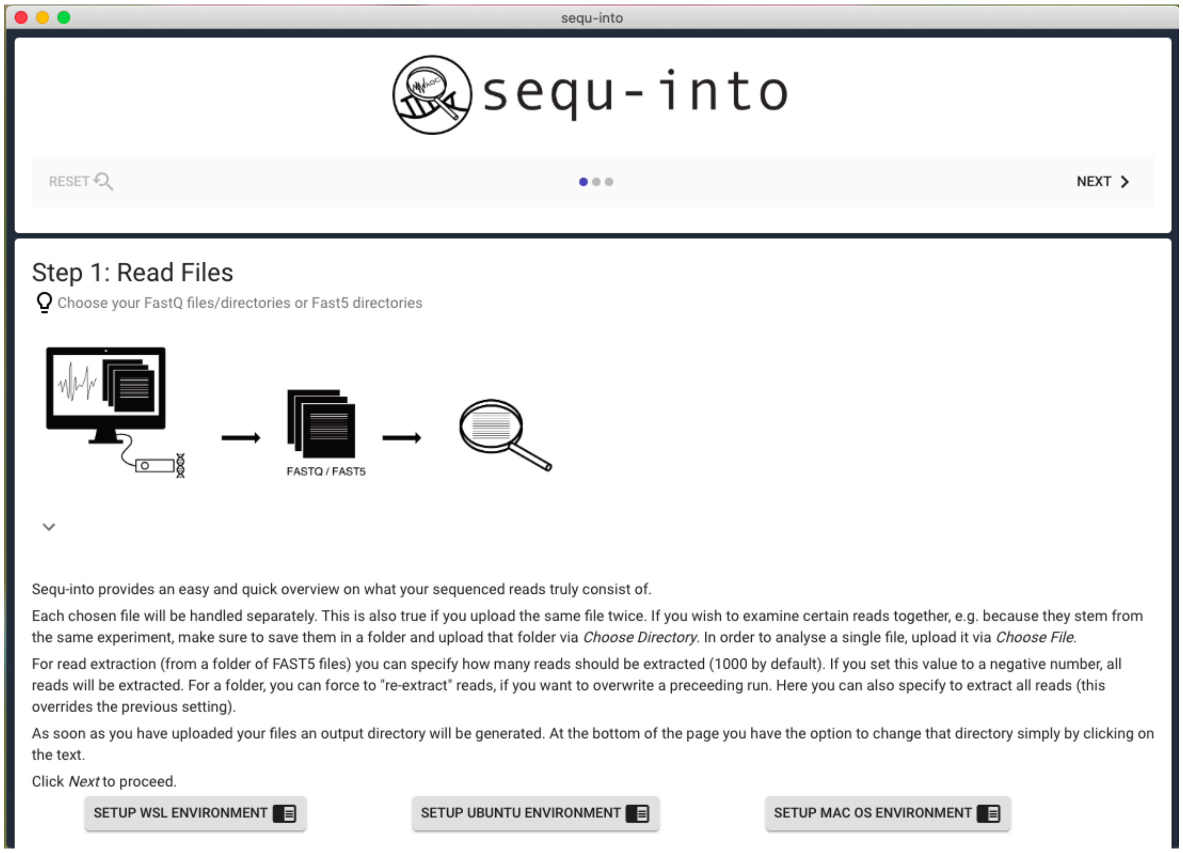
A DMG image will be downloaded. Open the image and drag the sequ-into app into your applications folder.



You are now ready to start sequ-into. It might be necessary to convince Mac OS to run this app for security reasons. We have created a brief tutorial which shows you how you can achieve this: *Overriding Mac OS security settings* .

2.1.3 Install dependencies

The authors of sequ-into prepared a shell script to automatically install the required python dependencies. You can access it from the first step description menu:



Step 1: Read Files

Choose your FastQ files/directories or Fast5 directories

Sequ-into provides an easy and quick overview on what your sequenced reads truly consist of.

Each chosen file will be handled separately. This is also true if you upload the same file twice. If you wish to examine certain reads together, e.g. because they stem from the same experiment, make sure to save them in a folder and upload that folder via *Choose Directory*. In order to analyse a single file, upload it via *Choose File*.

For read extraction (from a folder of FAST5 files) you can specify how many reads should be extracted (1000 by default). If you set this value to a negative number, all reads will be extracted. For a folder, you can force to "re-extract" reads, if you want to overwrite a preceeding run. Here you can also specify to extract all reads (this overrides the previous setting).

As soon as you have uploaded your files an output directory will be generated. At the bottom of the page you have the option to change that directory simply by clicking on the text.

Click *Next* to proceed.

SETUP WSL ENVIRONMENT SETUP UBUNTU ENVIRONMENT SETUP MAC OS ENVIRONMENT

Click on the *Setup MAC OS Environment* button and enter your user account password (you must be administrator). Beware, that due to technical limitations, your password will be shown on the console once.

2.2 Windows

Since Microsoft Windows is the only not POSIX based operating system supported by sequ-into, a little more action must be taken.

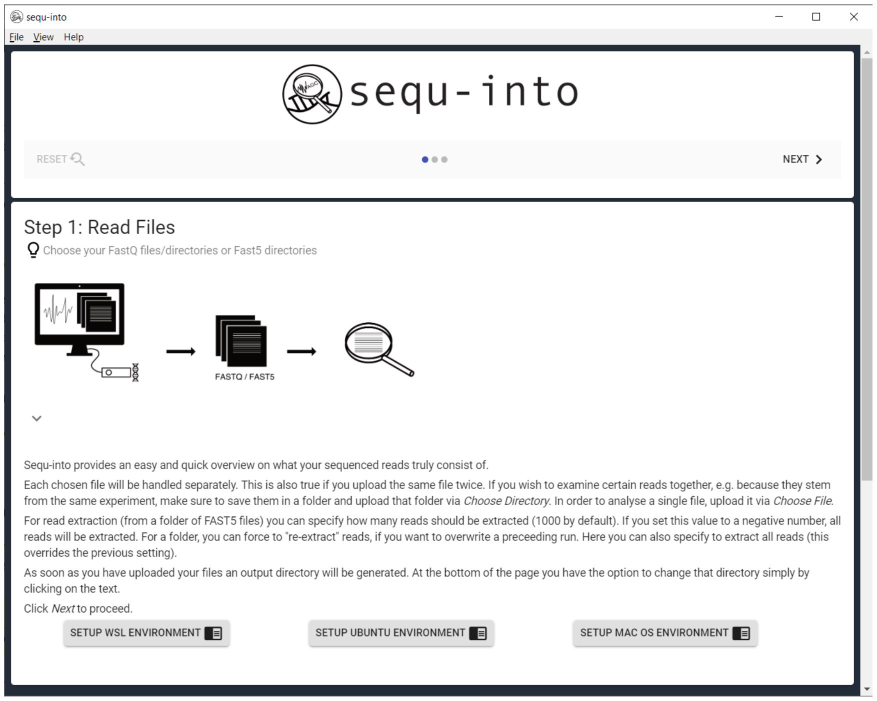
If you have not yet installed Windows Subsystem for Linux (also known as WSL/Bash on Ubuntu/Ubuntu/. . .) please do so. We have prepared a guide on how to do so in the chapter *How to setup Windows Subsystem for Linux* .

2.2.1 Installing Packages into WSL

After you have installed WSL, we must install some dependencies that are needed by our application. Please note, the following guide is aimed at having an Ubuntu installed. However, if you have installed a different distribution, we are sure you know what you are doing and hence, you don't need detailed help ;) .

2.2.2 Package Installation (Automatic)

The easist way to install all dependencies is to expand the description on the first step and click the *Setup WSL environment* button:



Step 1: Read Files

Choose your FastQ files/directories or Fast5 directories

Sequ-into provides an easy and quick overview on what your sequenced reads truly consist of.

Each chosen file will be handled separately. This is also true if you upload the same file twice. If you wish to examine certain reads together, e.g. because they stem from the same experiment, make sure to save them in a folder and upload that folder via *Choose Directory*. In order to analyse a single file, upload it via *Choose File*.

For read extraction (from a folder of FAST5 files) you can specify how many reads should be extracted (1000 by default). If you set this value to a negative number, all reads will be extracted. For a folder, you can force to "re-extract" reads, if you want to overwrite a preceeding run. Here you can also specify to extract all reads (this overrides the previous setting).

As soon as you have uploaded your files an output directory will be generated. At the bottom of the page you have the option to change that directory simply by clicking on the text.

Click *Next* to proceed.

This starts a *cmd* script which asks you for your *WSL* password and will execute the below steps automatically.

2.2.3 Package Installation (Manual)

Unfortunately sequ-into depends on several smaller libraries and applications, which we now have to install into WSL.

```
sudo apt-get update
sudo apt-get install git build-essential python3 python3-pip hdf5-tools libhdf5-
↪serial-dev
sudo pip3 install mappy matplotlib h5py flask pandas upsetplot
```

You will be asked to enter your *WSL* password when you submit your first *sudo* command. However, since *sudo* will give you administrator right in *WSL*, it might also be that it asks for your password everytime.

The following will explain the packages and software going to be installed. Since you provided your *sudo*-password, you should get to know what we are doing ;) If you are not interested: congratulations, you're done!

First a basic developer package has to be installed, which is done by installing *git* for version control/access to repositories, *build-essential* to get C/C++ compilers (to build other software) and python3 for running the online-/incremental algorithm, generating reads from fast5 files and making the statistics. *python3-pip* is the python package manager which is need to install further python packages. Finally libhdf5, hdf5-tools and h5py are needed to access fast5 files. For alignment of the reads we rely on mappy/minimap2 . matplotlib and upsetplot is needed to prepare plots. pandas is used to prepare data for the plots and flask is used to run the server for the incremental updates.

After you have completed these steps, you are ready to use sequ-into!

2.2.4 Executable

We have built sequ-into as a portable app. You thus only need to place the downloaded executable at any location and can start using it (after you have setup *WSL* once on your computer).

2.3 Linux/Source

We are not providing a binary download for Linux, since we assume that you are familiar with the command line, if your computer runs Linux. In explanation on why a software is needed can be found above at _wslpackinstall .

First you must clone the sequ-into repository , install with npm and finally build our tool.

```
git clone https://github.com/mjoppich/igem_munich_2018.git
cd igem_munich_2018
npm install

npm run build
npm package-linux
```

You will find the sequ-into application in igem_munich_2018/release/ .

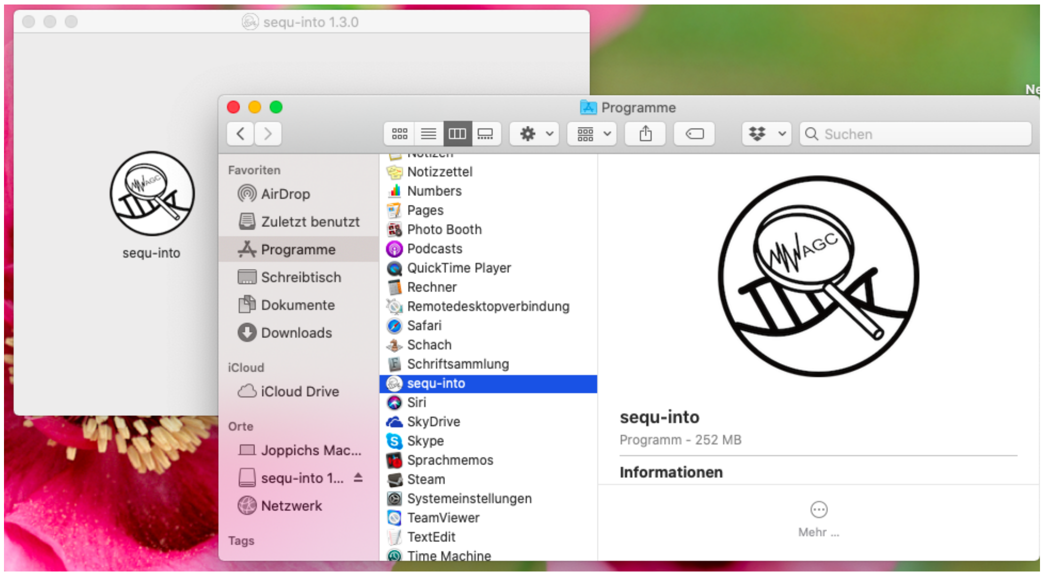
In order to have all python scripts running, please install the following dependencies. You may leave *python3-pip* out if you are using your own pip or anaconda.

```
sudo apt-get update
sudo apt-get install git build-essential python3 python3-pip hdf5-tools libhdf5-
↪serial-dev
sudo pip3 install mappy matplotlib h5py flask pandas upsetplot
```

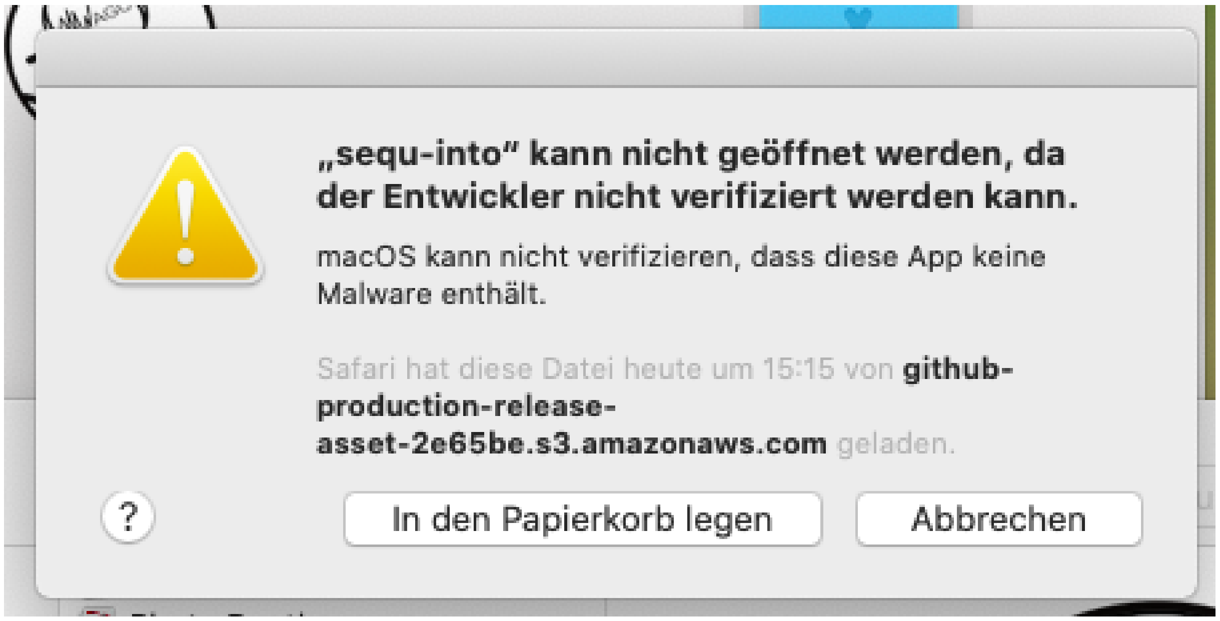
2.4 Overriding Mac OS security settings

In order to execute non-App-store apps on Mac OS, you must force your Mac to do so (or pay the authors some money, so we can afford the registration :)).

Upon downloading and dragging sequ-into into the Apps folder on your Mac (Programme folder in the screenshots), you can double-click on the sequ-into app.



A popup will be shown, telling you that you download sequ-into from some remote location and that the app is not signed. Hence you can either cancel or move the app into the bin.

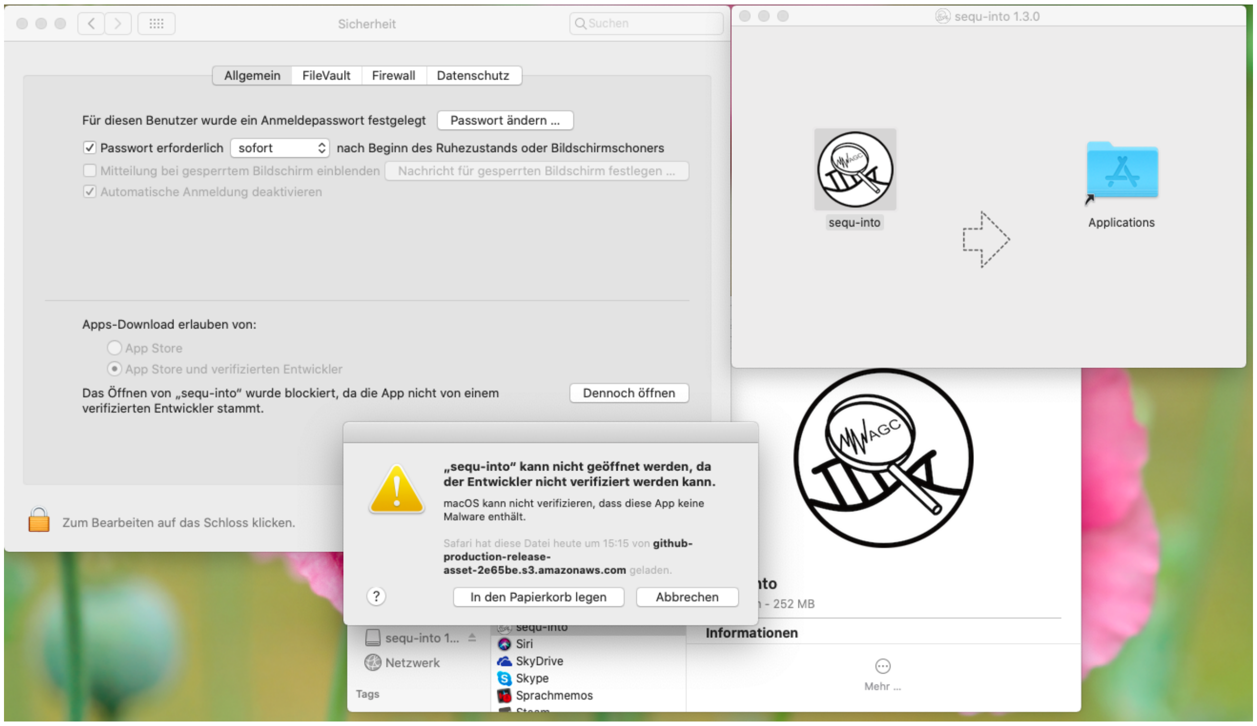


„sequ-into" kann nicht geöffnet werden, da der Entwickler nicht verifiziert werden kann.

macOS kann nicht verifizieren, dass diese App keine Malware enthält.

Safari hat diese Datei heute um 15:15 von **github-production-release-asset-2e65be.s3.amazonaws.com** geladen.

In order to allow the execution of sequ-into, please go into your Settings -> Security -> General. Verify that you allow to download apps from the app store and verified developers. Since you already tried to execute sequ-into once, the general page will also tell you, that the execution of the *sequ-into* app has been blocked, because it has not been created by a verified developer. A button enables you to open sequ-into anyway. Click on that button.



Für diesen Benutzer wurde ein Anmeldepasswort festgelegt [Passwort ändern ...]

[✓] Passwort erforderlich [sofort] nach Beginn des Ruhezustands oder Bildschirmschoners

[ ] Mitteilung bei gesperrtem Bildschirm einblenden [Nachricht für gesperrten Bildschirm festlegen ...]

[✓] Automatische Anmeldung deaktivieren

Apps-Download erlauben von:

App Store [✓ App Store und verifizierten Entwickler]

Das Öffnen von „sequ-into" wurde blockiert, da die App nicht von einem verifizierten Entwickler stammt. [Dennoch öffnen]

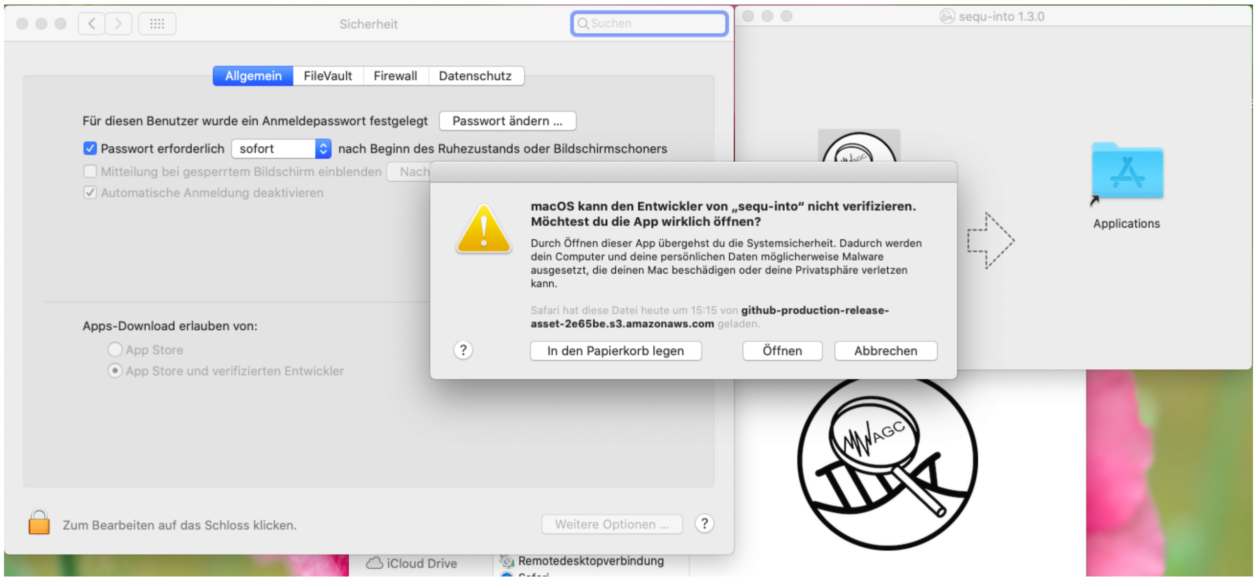
Zum Bearbeiten auf das Schloss klicken.

„sequ-into" kann nicht geöffnet werden, da der Entwickler nicht verifiziert werden kann.

macOS kann nicht verifizieren, dass diese App keine Malware enthält.

Safari hat diese Datei heute um 15:15 von **github-production-release-asset-2e65be.s3.amazonaws.com** geladen.

After hitting the button to open sequ-into anyway, another popup will be shown. It will again tell you that Mac OS does not know who created the app, and that this app is potentially unsafe. This time, it also allows you to *Open* the app. Click on *Open*.



Für diesen Benutzer wurde ein Anmeldepasswort festgelegt

☑ Passwort erforderlich sofort nach Beginn des Ruhezustands oder Bildschirmschoners

☐ Mitteilung bei gesperrtem Bildschirm einblenden

☑ Automatische Anmeldung deaktivieren

Apps-Download erlauben von:

○ App Store

● App Store und verifizierten Entwickler

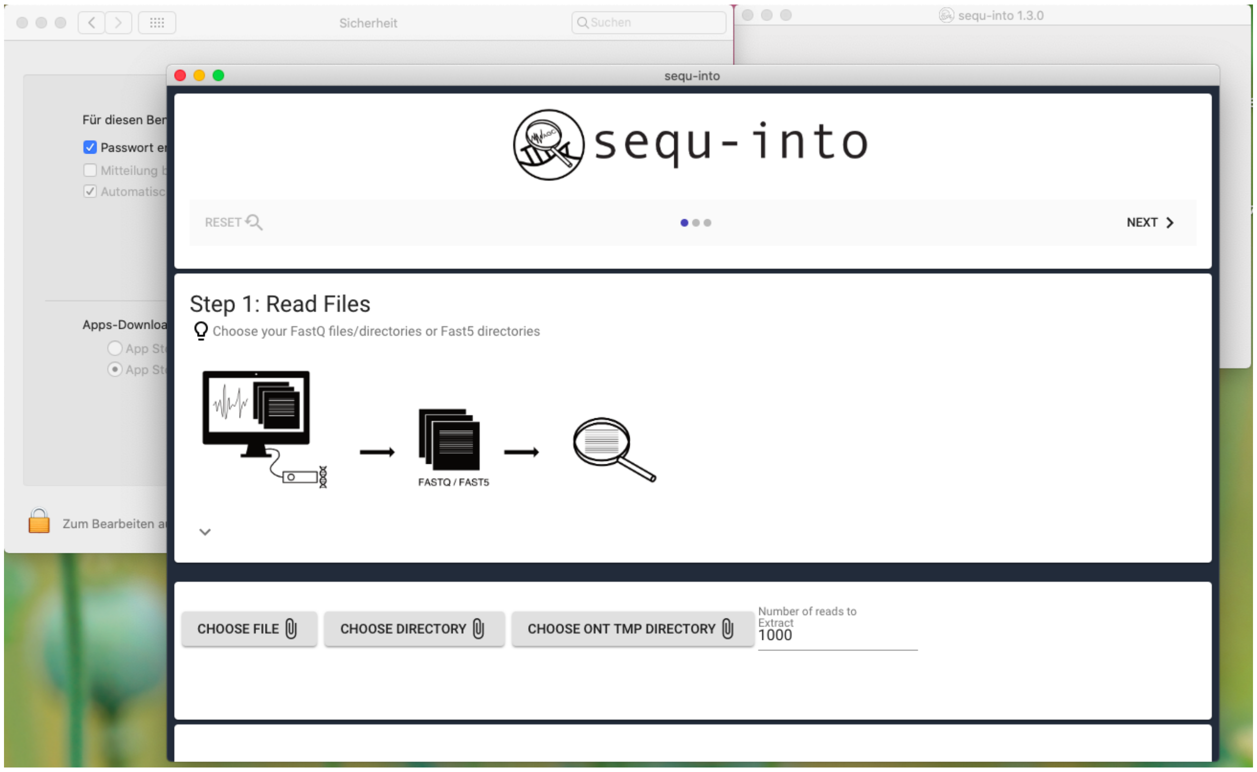
Zum Bearbeiten auf das Schloss klicken.

macOS kann den Entwickler von „sequ-into“ nicht verifizieren. Möchtest du die App wirklich öffnen?

Durch Öffnen dieser App übergehst du die Systemsicherheit. Dadurch werden dein Computer und deine persönlichen Daten möglicherweise Malware ausgesetzt, die deinen Mac beschädigen oder deine Privatsphäre verletzen kann.

Safari hat diese Datei heute um 15:15 von **github-production-release-asset-2e65be.s3.amazonaws.com** geladen.

After all these steps, we could finally convince Mac OS to run sequ-into. Congratulations!



Step 1: Read Files

Choose your FastQ files/directories or Fast5 directories

Number of reads to Extract: 1000

CHAPTER 3

How to setup Windows Subsystem for Linux

Depending on your Windows version you need to activate Developer Mode first. How this is done is explained at the end of this page, since this is only required for old versions of Windows 10.

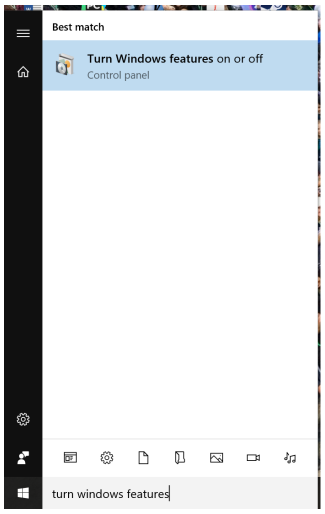
For all recent versions of Windows 10 you can start with step 1.

Warning: Some antivirus software (e.g. Kaspersky) disable internet access for unknown/new programs. Make sure you have internet access!

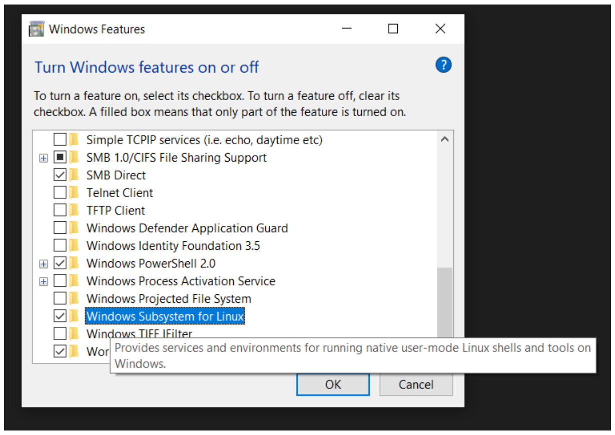
On newer Windows systems, step 1 might not be needed.

3.1 Step 1: Activate WSL feature

First WSL has to be enabled from Windows features. Therefore, simply search for the *Turn Windows features on or off* option in the control panel.



Once found, look for the *Windows Subsystem for Linux (Beta)* row and make sure to check the corresponding box.


Windows Features

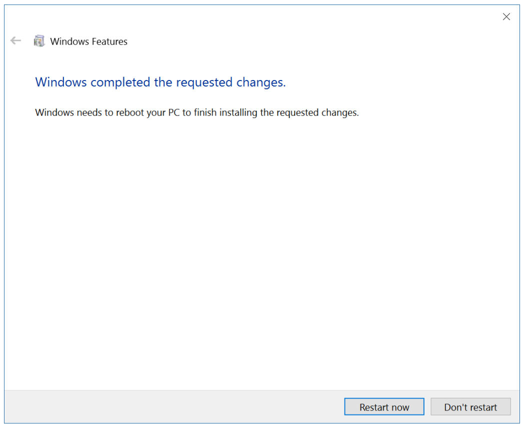
Turn Windows features on or off

To turn a feature on, select its checkbox. To turn a feature off, clear its checkbox. A filled box means that only part of the feature is turned on.

- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ■ SMB 1.0/CIFS File Sharing Support
- ☑ SMB Direct
- ☐ Telnet Client
- ☐ TFTP Client
- ☐ Windows Defender Application Guard
- ☐ Windows Identity Foundation 3.5
- ☑ Windows PowerShell 2.0
- ☐ Windows Process Activation Service
- ☐ Windows Projected File System
- ☑ Windows Subsystem for Linux
- ☐ Windows TIFF IFilter
- ☑ Wor

Provides services and environments for running native user-mode Linux shells and tools on Windows.

This will download and install the desired WSL feature. Finally apply the change and make sure to reboot your computer


Windows Features

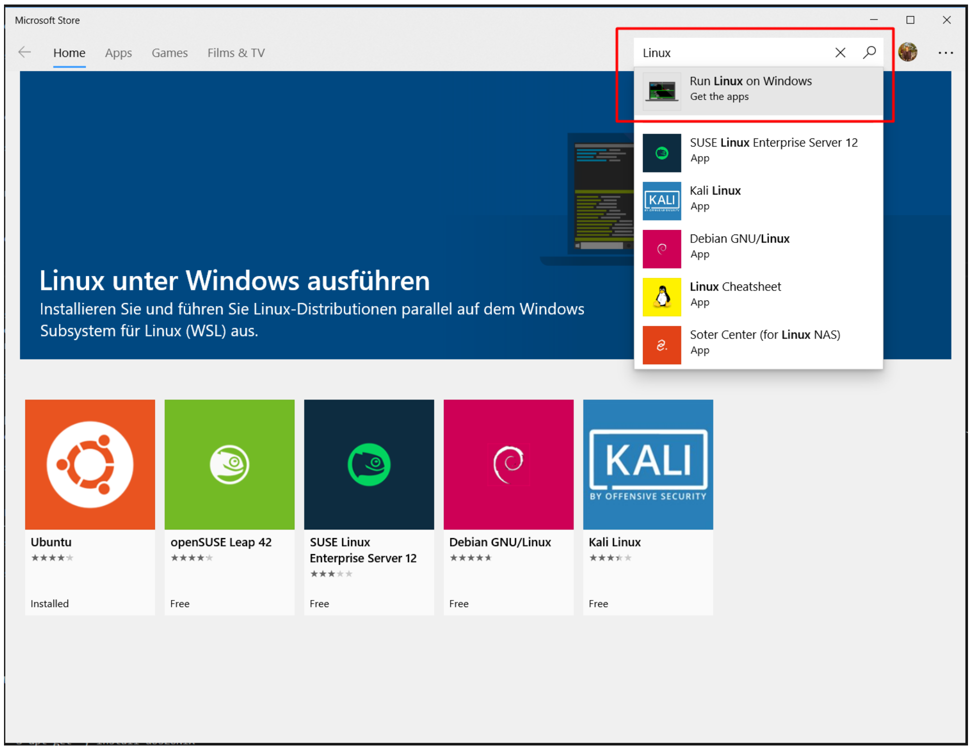
Windows completed the requested changes.

Windows needs to reboot your PC to finish installing the requested changes.

3.2 Step 2: Install Linux

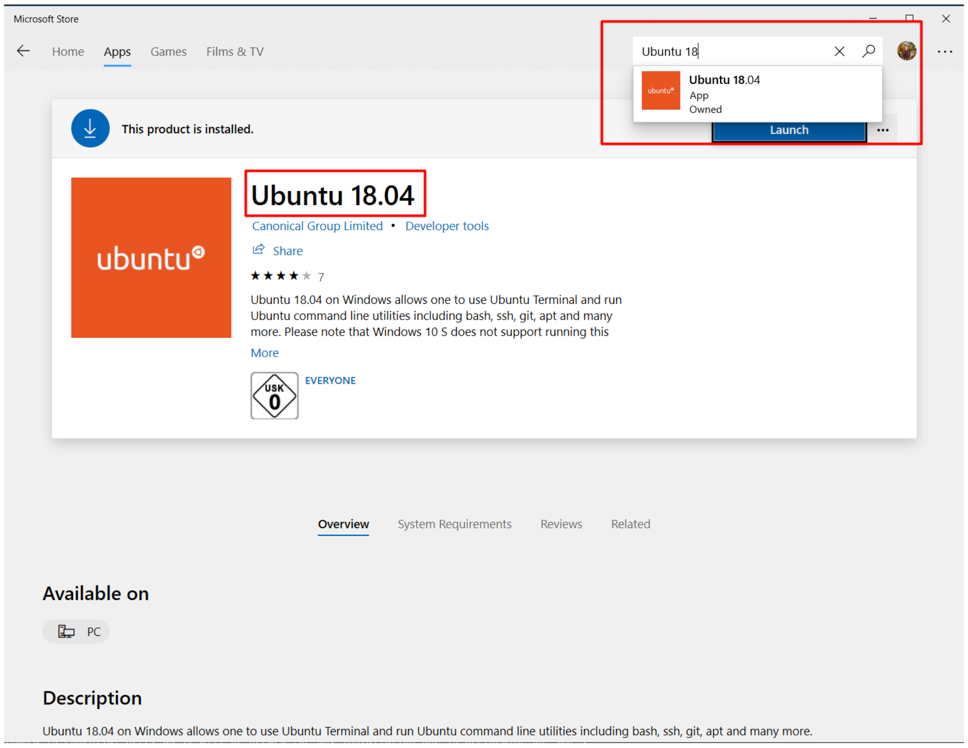
After having enabled the WSL feature, we can visit the Microsoft Windows Store to download Linux.

In order to do so, we open the Windows Store app, and search for *Linux*. We select the *Run Linux on Windows* menu entry.


Microsoft Store
Home
Apps
Games
Films & TV
Linux
Run Linux on Windows
Get the apps
SUSE Linux Enterprise Server 12
App
Kali Linux
App
Debian GNU/Linux
App
Linux Cheatsheet
App
Soter Center (for Linux NAS)
App
Linux unter Windows ausführen
Installieren Sie und führen Sie Linux-Distributionen parallel auf dem Windows Subsystem für Linux (WSL) aus.
Ubuntu
Installed
openSUSE Leap 42
Free
SUSE Linux Enterprise Server 12
Free
Debian GNU/Linux
Free
Kali Linux
Free

There are many different flavors (comparable to strains in biology) of linux and some are already offered on the Windows store. Best compatibility for *sequ-into* has Ubuntu.

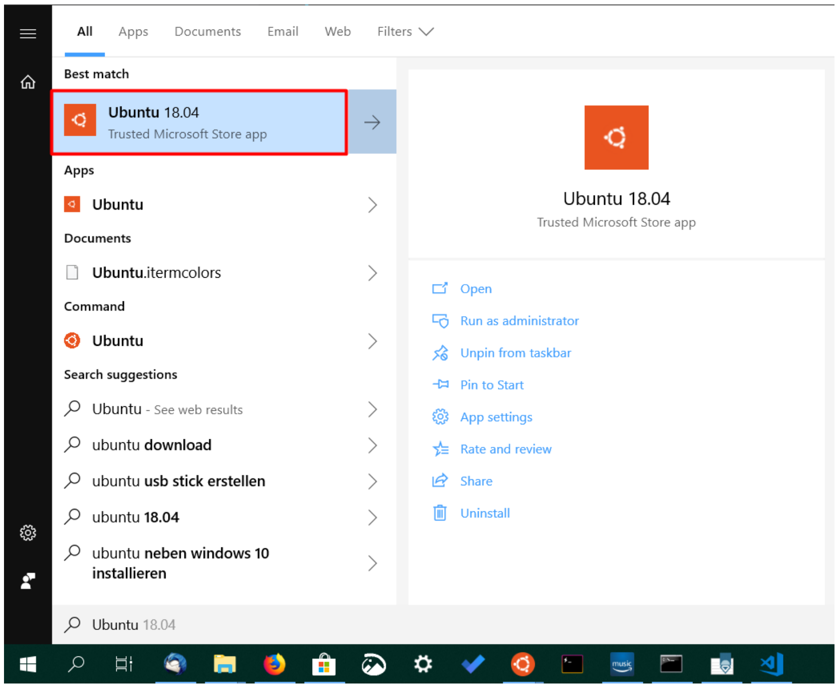
Important: You should consider using the latest Ubuntu version available. This is Ubuntu 18.04 at the time of writing. You specifically have to search for *Ubuntu 18.04* in the store !



Ubuntu 18.04 on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more. Please note that Windows 10 S does not support running this

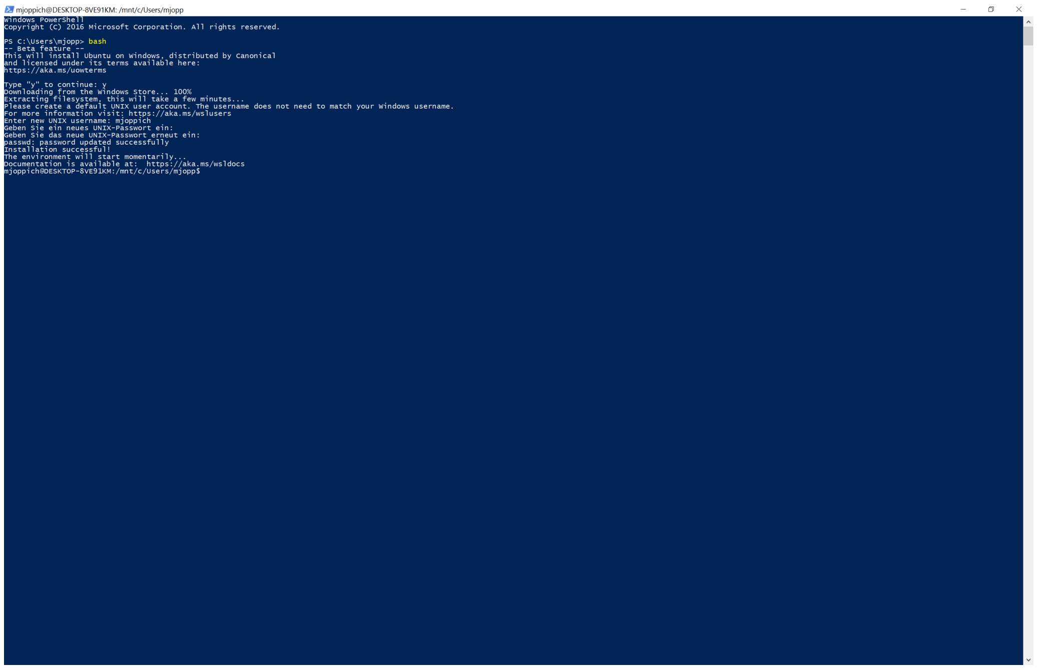
Ubuntu 18.04 on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

Now let the Windows store install your Linux app and once that is done, open your newly installed Linux:



The black screen will guide you through the install process. It will first unpack itself and then ask you to create a linux user account.

It is recommended to choose a username and password you can easily remember. Remembering the password is essential here, as it will be needed for any installation to be performed on *WSL*!


mjoppich@DESKTOP-8VE91KM: /mnt/c/Users/mjopp

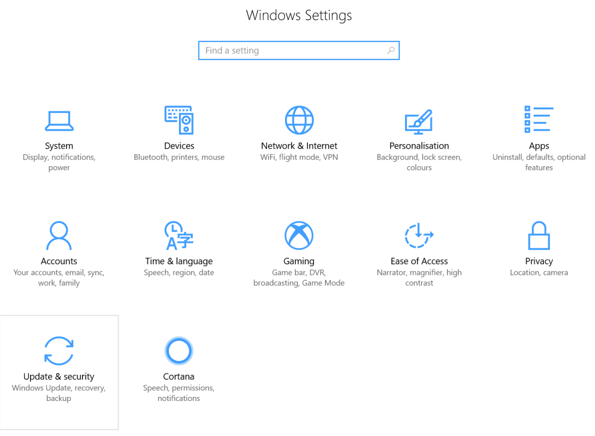
```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\mjopp> bash
-- Beta feature --
This will install Ubuntu on Windows, distributed by Canonical
and licensed under its terms available here:
https://aka.ms/uowterms

Type "y" to continue: y
Downloading from the Windows Store... 100%
Extracting filesystem, this will take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: mjoppich
Geben Sie ein neues UNIX-Passwort ein:
Geben Sie das neue UNIX-Passwort erneut ein:
passwd: password updated successfully
Installation successful!
The environment will start momentarily...
Documentation is available at:  https://aka.ms/wsldocs
mjoppich@DESKTOP-8VE91KM:/mnt/c/Users/mjopp$
```

3.3 Step 3: Prepare WSL

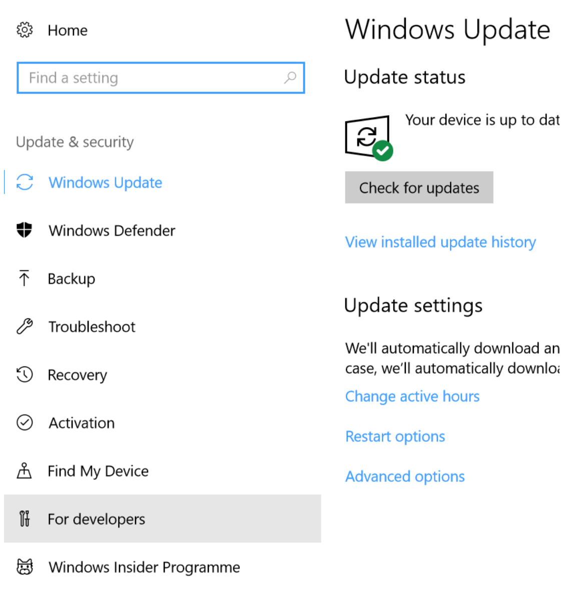
Before you can use *sequ-into* on *WSL*/Ubuntu please make sure to follow the *Package Installation (Automatic)* instructions.

Step 0: Activate Developer Mode

Since WSL/Bash on Ubuntu on Windows is a developer feature, first the developer mode has to be actived. Therefore we go into the Settings app and select *Update & Security*.


Windows Settings
Find a setting
System
Display, notifications, power
Devices
Bluetooth, printers, mouse
Network & Internet
WiFi, flight mode, VPN
Personalisation
Background, lock screen, colours
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, family
Time & language
Speech, region, date
Gaming
Game bar, DVR, broadcasting, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Privacy
Location, camera
Update & security
Windows Update, recovery, backup
Cortana
Speech, permissions, notifications

We further navigate into the *For developers* tab on the left.



Windows Update

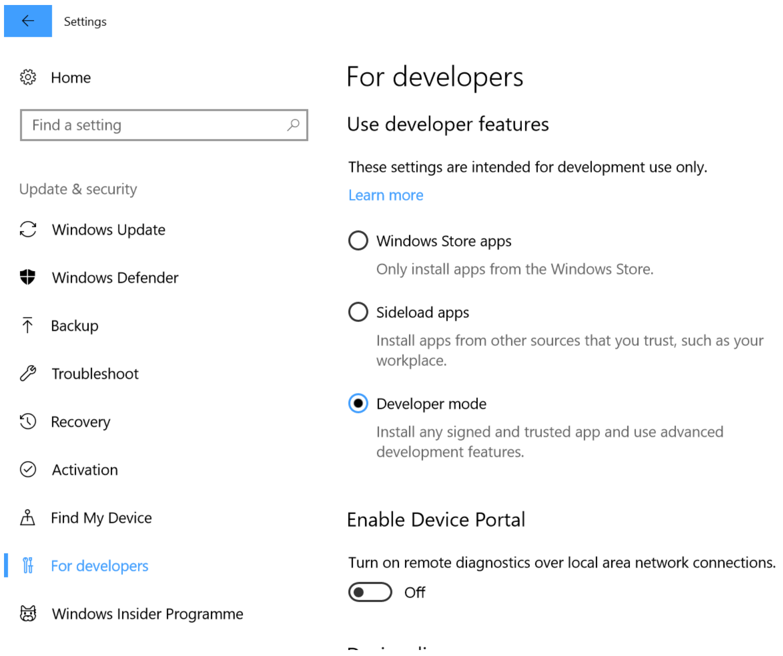
Update status

Your device is up to dat

Update settings

We'll automatically download an
case, we'll automatically downlo:

In the *For developers* options we switch from *Windows Store apps* to *Developer mode*.



For developers

Use developer features

These settings are intended for development use only.

○ Windows Store apps
Only install apps from the Windows Store.

○ Sideload apps
Install apps from other sources that you trust, such as your workplace.

● Developer mode
Install any signed and trusted app and use advanced development features.

Enable Device Portal

Turn on remote diagnostics over local area network connections.

Off

Warning: This setup guide is taken from bioGUI documentation from the original author for reasons.

CHAPTER 4

User Guide

4.1 How to get sequ-into?

You can use *sequ-into* on a Mac OS, Linux as well as on a Windows System. Please follow the respective instructions in our *Installation* guide.

4.2 Get started

4.2.1 Step 1: Read files

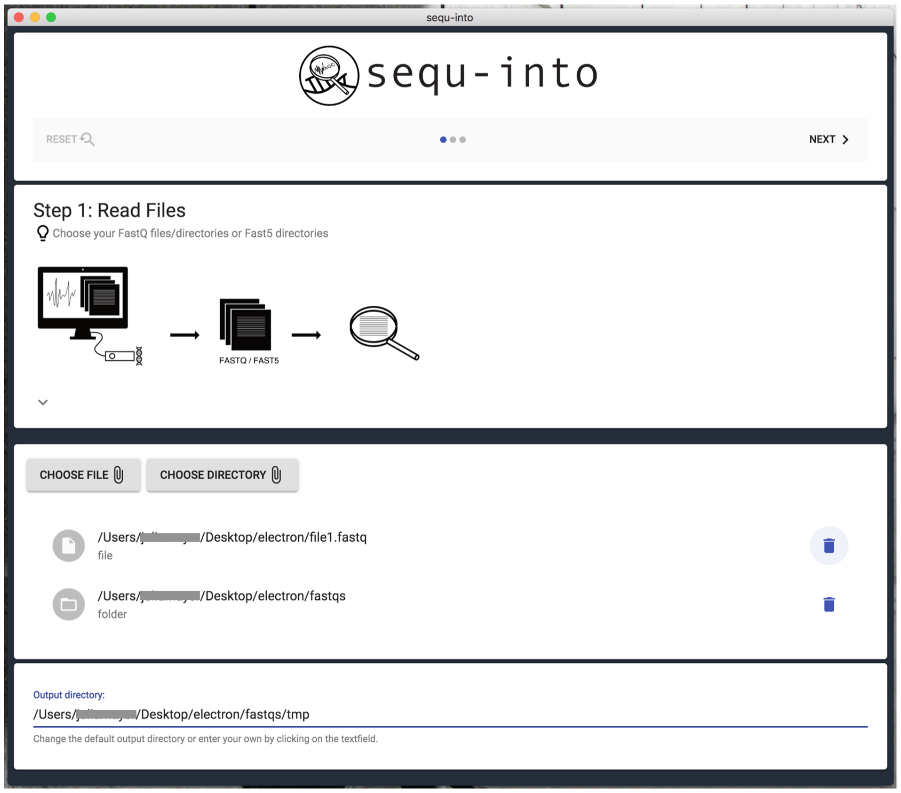
FastQ, as well as Fast5, are suitable formats for evaluating your sequencing data with *sequ-into*.

In the first step, you can choose which files you would like to seek into. Each chosen file or folder will be handled separately. This is also true if you upload them twice.

If you wish to examine certain reads together, e.g. because they stem from the same experiment, make sure to save them in a folder and upload that folder via *Choose Directory*. In order to analyze a single file, upload it via *Choose File*.

As soon as you have chosen your files an output directory will be generated. You will find a temp folder where your read files reside. You can change that output directory and folder name at the bottom of the page if you click on the text field.

After that, click *Next* to proceed.



Step 1: Read Files

Choose your FastQ files/directories or Fast5 directories

/Users/[redacted]/Desktop/electron/file1.fastq
file

/Users/[redacted]/Desktop/electron/fastqs
folder

Output directory:
/Users/[redacted]/Desktop/electron/fastqs/tmp
Change the default output directory or enter your own by clicking on the textfield.

4.2.2 Step 2: Reference files

To check what your sequencing files truly consist of you need a reference against which the reads will be mapped.

That reference might be a possible contamination, such as *E. Coli*, or a targeted known genome of what you intended to sequence. Of course, you can also use shorter sequences instead of a whole genome as a reference. For details on possible technical limitations, please see GraphMap and https://www.nature.com/articles/ncomms11307.

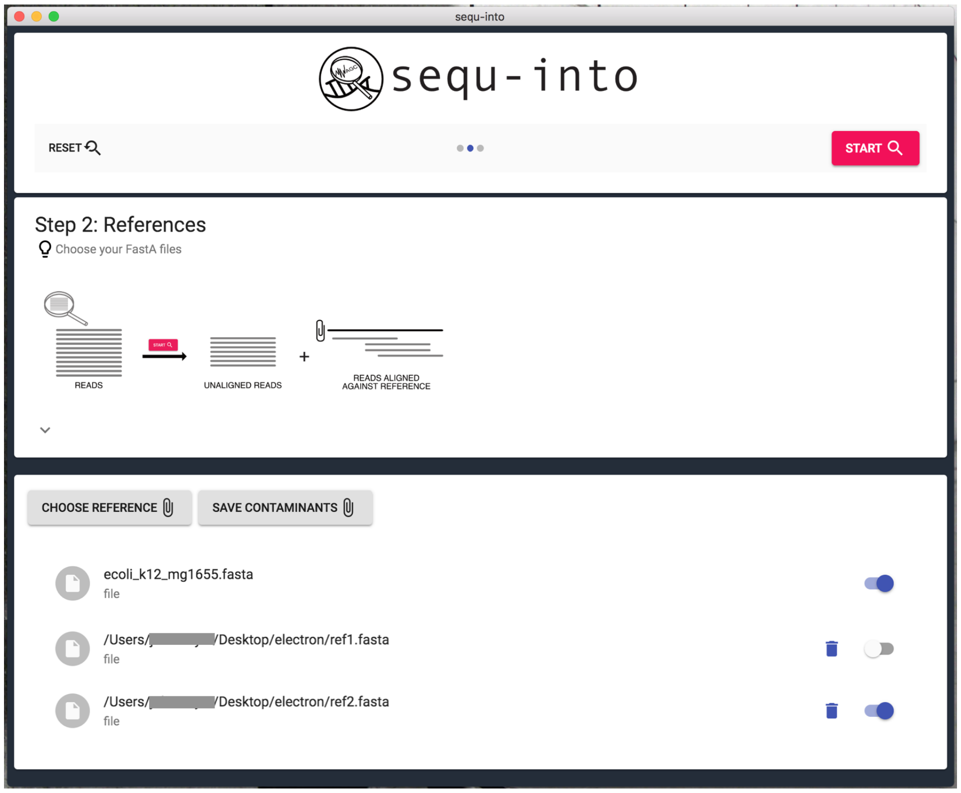
Mapping is possible against RNA as well as against DNA sequences, as long as they are in the FastA Format. You can find sequences for example on NCBI https://www.ncbi.nlm.nih.gov/genome/?term=.

Click on *Choose Reference* to choose your reference files. You can select as many files as you wish. These files will still be present after you used *Reset*, but are deleted when you close the application.

If you work with certain references repeatedly they can also be saved in the app so that they are available every time even after you closed *sequ-into*. Simply *Save Contaminants*. Your own references can always be deleted from *sequ-into* later on, just click the trash can to do so.

Keep in mind that calculation time increases with file size and file quantity! Consider using the switches behind each reference to turn them off if you don't need them for your current run. They will still be available after you used *Reset*.

After that, click *Start* to run the calculations.



Step 2: References

Choose your FastA files

ecoli_k12_mg1655.fasta
file

/Users/[redacted]/Desktop/electron/ref1.fasta
file

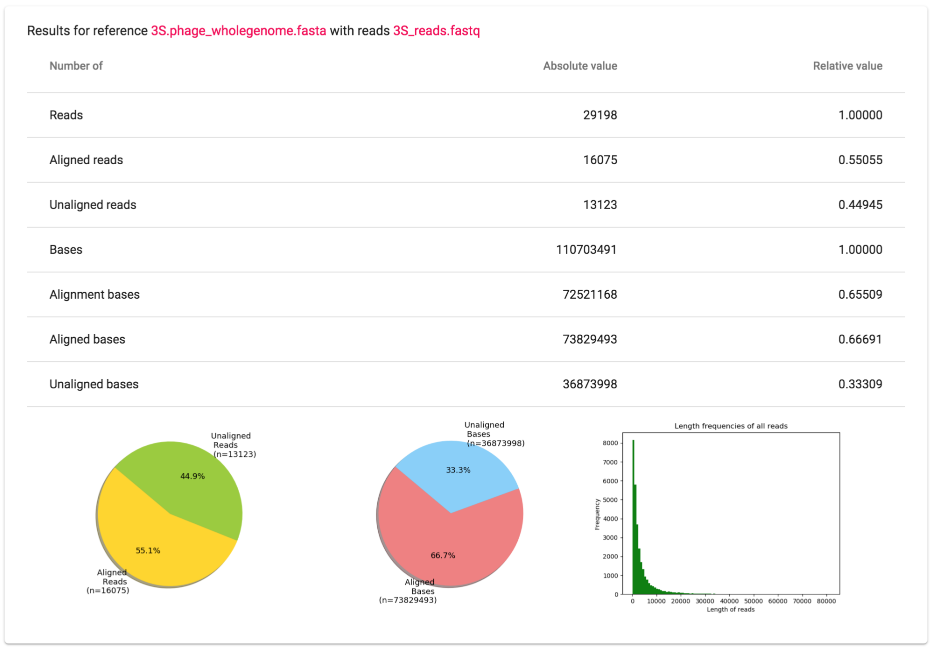
/Users/[redacted]/Desktop/electron/ref2.fasta
file

4.2.3 Step 3: Results

The Results consist of two sections: a statistical overview on how your reads mapped to the reference(s) and the filter to extract and save only those reads you need for your downstream analysis.

4.2.4 Section 1:

For each combination of FastQ (file/directory) with FastA you will find one table and three plots.



Results for reference 3S.phage_wholegenome.fasta with reads 3S_reads.fastq

Number of	Absolute value	Relative value
Reads	29198	1.00000
Aligned reads	16075	0.55055
Unaligned reads	13123	0.44945
Bases	110703491	1.00000
Alignment bases	72521168	0.65509
Aligned bases	73829493	0.66691
Unaligned bases	36873998	0.33309

Unaligned Reads (n=13123)
44.9%
55.1%
Aligned Reads (n=16075)
Unaligned Bases (n=36873998)
33.3%
66.7%
Aligned Bases (n=73829493)
Length frequencies of all reads
Frequency
Length of reads

The table includes read and base frequencies in the reference FastA file. For reads, you receive the information about aligned or not aligned reads. It is not always sufficient enough to rely only on reads in the further analysis. The different read sizes can cause the wrong interpretation of the data: three contaminated reads of length 50 bp or 5000 bp make a big difference despite the fact that there is three of them in both cases. For making proper conclusions about the data it is useful to take a look on the bases as well. For bases, it is important to note that there are two different definitions: *alignment bases* and *aligned bases*.

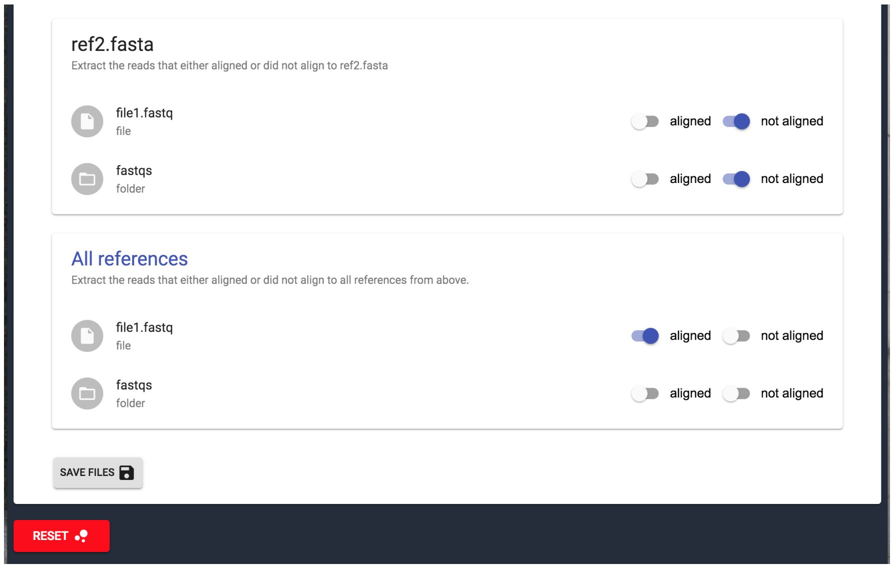
Aligned reads consist out of bases. These bases are called the *aligned bases*. On the other hand, the bases that are indeed aligned, means mapped to the base in the reference and are not skipped, are called *alignment bases*.

To support the statistical information in the table visually we also added two pie charts that correspond to the relative and absolute values in the table. These two plots will help you to gain information about the number of bases and reads that were found in a reference file and make a conclusion about the possibility of contamination.

Additionally, there is a bar plot representing the distribution of the read length in the FastQ file you uploaded. This chart could be used for evaluation of the quality of sequencing or even be helpful by evolving theories about files with filtered reads. For your onvenience **all plots are saved in the output directory** specified in Step 1.

4.2.5 Section 2:

In the section below you will find a filter which you can optionally use to extract and save distinguish parts of the read FastQ file: reads that were mapped to the reference (*aligned switch*) and those which were not (*not aligned switch*), in other words possibly contaminated reads and reads that can be used for downstream analysis (in case the reference FastA file you used is a possible contaminant. If you added the FastA file of the organism you expect to sequence, *not aligned* reads are contamination).


ref2.fasta
Extract the reads that either aligned or did not align to ref2.fasta

file1.fastq (file)	aligned	[✓ not aligned]
fastqs (folder)	aligned	[✓ not aligned]

All references
Extract the reads that either aligned or did not align to all references from above.

file1.fastq (file)	[✓ aligned]	not aligned
fastqs (folder)	aligned	not aligned

If you uploaded multiple references files one more filter will appear (*All references*): filter of reads that are aligned to **all** references or reads that are aligned to **none** of the references.

With this filter, it is possible to refine sequencing data and consequently, achieve preferable results by downstream analysis. It can also give you a hint about the origin of the possible contamination, as the reads that are not mapped to the expected organism can be checked with BLAST.

Once again all files will be saved in your output directory specified in Step 1.

CHAPTER 5

How does Sequ-Into work?

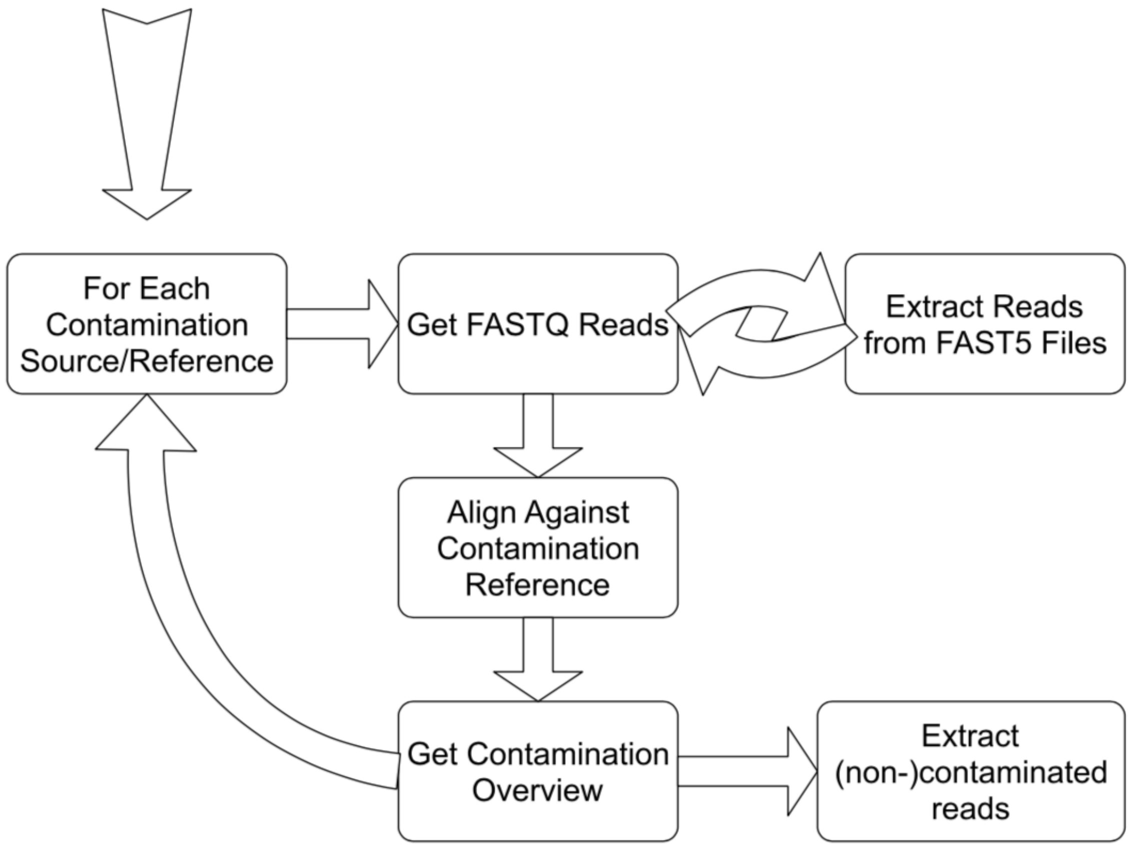
sequ-into has the aim of bringing the sequencing data analysis and the laboratory protocol optimization in close proximity.

While highly specialized tools and pipelines for third generation sequencing data analysis are available, they often are not handy nor convenient to use as a first assessment right after or during the sequencing run.

As a possible solution we brought together a straightforward intuitive interface built with Electron and React, that gives the user easy access to the state-of-the-art long read alignment tool Minimap2/mappy which itself is highly specialized for nanopore sequencing.

To make this possible we run a python script in the background that relies on HTSeq as infrastructure for high-throughput data and pysam to handle the genomic data sets.

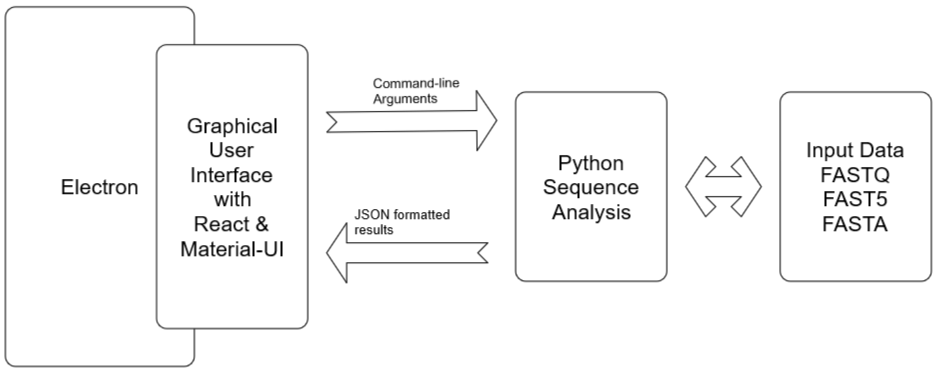
5.1 What does sequ-into do?


For Each Contamination Source/Reference
Get FASTQ Reads
Extract Reads from FAST5 Files
Align Against Contamination Reference
Get Contamination Overview
Extract (non-)contaminated reads

In order to be able to draw conclusions of the sequencing quality in general and the composition of the data - in terms of contaminations versus the true sequencing traget - the reads are mapped to references. The reference being either a possible contamination, leaving your desired reads unaligned, or your target sequence, meaning your designated reads are the ones that did align. The distribution of read length from the original files and the results of these alignments are then elucidated in a statistical overview and employed to separate those reads you aimed for from those that were sequenced involuntary.

5.2 How does sequ-into achieve this?

5.2.1 From a Typescript interface to functionality


Electron
Graphical User Interface with React & Material-UI
Command-line Arguments
JSON formatted results
Python Sequence Analysis
Input Data FASTQ FAST5 FASTA

The user interface of *sequ-into* is based on Electron and React and written in Typescript. However, the functionality of our app depends on a python script (*ContamTool.py*) in the background, that must be called according to the users request.

Read Files

sequ-into is able to deal with both, the FastQ as well as the Fast5 format. If the latter is used, we extract the base called sequences and convert them into the FastQ format.

Thanks to the fact that the Fast5 format is in fact HDF5, a file format that can contain an unlimited variety of datatypes while allowing for input/output of complex data, it was possible to manipulate the files with the h5py python interface efficiently. To prevent excessive runtimes of our app, there is currently a processing limit of 1000 reads per Fast5 file.

```
return OrderedDict([
    (Fast5TYPE.BASECALL_2D, '/Analyses/Basecall_2D_%03d/'),
    (Fast5TYPE.BASECALL_1D_COMPL, '/Analyses/Basecall_1D_%03d/'),
    (Fast5TYPE.BASECALL_1D, '/Analyses/Basecall_1D_%03d/'),
    (Fast5TYPE.BASECALL_RNN_1D, '/Analyses/Basecall_RNN_1D_%03d/'),
    (Fast5TYPE.BARCODING, '/Analyses/Barcoding_%03d/'),
    (Fast5TYPE.PRE_BASECALL, '/Analyses/EventDetection_%03d/')
])
```

After acquiring the sequenced data meant to be analyzed, *sequ-into* handles each uploaded file/folder as a separated call. In the case of a folder, *sequ-into* searches for each file in that directory down to the deepest level of the directory tree.

```
self.state.inputFiles.forEach(element => {

        var stats = fs.lstatSync(element.path)

        if (stats.isDirectory()){
            var allFilesInDir = fs.readdirSync(element.path);
            processFilesForElement[element.path] = [];

            allFilesInDir.forEach((myFile:any) => {
                if(myFile.toUpperCase().endsWith("FASTQ") || myFile.toUpperCase().
↪endsWith("FQ")){
                    var pathToFile = self.normalizePath(path.join(element.path,␣
↪myFile));
```

(continues on next page)

(continued from previous page)

```
                    processFilesForElement[element.path].push(pathToFile)
                }
            });

            if (processFilesForElement[element.path].length == 0){
                self.extractReadsForFolder(element.path);
            }
        }else{
            processFilesForElement[element.path] = [self.normalizePath(element.path)];
        }
    });
```

All files that are pooled in a folder are handled as one file in the further steps (*ContamTool.py*), resulting in a combined analysis of all the files in that folder.

Reference Files

The next step is to acquire the FastA files that are used as a reference for the alignment. As the user might have similar requests repeatedly, it is possible to save reference files in the app itself. To make these files available even after the app is closed, we use a JSON file to store their paths internally together with our default genome of *Escherichia coli* K-12 MG1655.

Cross Plattform Compatibility

Now that the required data is accessible, the python script (*ContamTool.py*) handling the alignment, calculation and plotting can be called.

As the *alignment-tool* we employed in our python script runs asynchron but since we have to make several calls for the functionality of *sequ-into*, one for each file per reference, we call the python script sequential.

```
child = spawnSync(
    program,
    programArgs,
        {
            cwd: process.cwd(),
            env: process.env,
            stdio: 'pipe',
            encoding: 'utf-8',
            shell: useShell
        })
```

To facilitate this on every platform *sequ-into* formulates the call command accordingly.

For a Unix system, this is simply:

```
var splitted_command = command.split(" ");
program = "python3";
programArgs = splitted_command;
useShell = true;
```

For Mac OS, the explicit path to all files is needed additionally:

```
var np = shellPath.sync();
process.env.PATH = np;
```

On Windows, however, it is necessary to make the call WSL compatible:

```
var splitCmd = ["-i", "-c", "python3 " + command];
program = "bash";
programArgs = splitCmd;
useShell = false;
```

Script Output

The output of each python call - that is for each file per reference - is collected via another JSON file data structure. More details *here*.

5.2.2 ContamTool.py

As mentioned above the functionality of *sequ-into* depends on the python script ContamTool.py which assesses the input read files, coordinates the alignment, interprets the alignment results and allows for read extraction according to the gained knowledge.

Read File Handling

All files that are pooled in a folder are handled as one FastQ file in the further steps to make the combined analysis possible.

```
fastqFile = os.path.join(output_dir, prefix + "complete.fastq")
os.system("cat " + ' '.join(read_file) + " > " + fastqFile)
```

Using the Alignment Tool Minimap2

The idea behind *sequ-into* that enables finding possible contaminations and deciding if a certain target was sequenced, respectively, is to map the raw reads from the sequencing files against a reference. Thus allowing to split the original joint read file into two categories: the reads that aligned to the reference and those that did not.

Nanopore sequencing data, however, comes with certain obstacles that complicate alignments. On the one hand, because of Nanopores high-throughput nature, the data size means that alignment algorithms commonly used are too slow - something that was overcome only with a tradeoff to lower sensitivity. On the other hand, the variable error profile of ONT MinION sequencers made parameter tuning mandatory to gain high sensitivity and precision. What makes *sequ-into* a reliable tool nevertheless, is Minimap2. This mapping algorithm is specifically designed to analyse long-read sequencing data, while it handles potentially high-error rates robustly and aligns long reads with speed and high precision thanks to a fast graph traversal. (Minimap2: pairwise alignment for nucleotide sequences, Heng Li, Bioinformatics, Volume 34, Issue 18, 15 September 2018, Pages 3094–3100,)

For each reference, Minimap2 is called with the input read file, generating a Sequence Alignment Map.

```
for refFileIdx, refFile in enumerate(cont_file):

a = mp.Aligner(refFile)  # load or build index
if not a:
    raise Exception("ERROR: failed to load/build index")
...
for fastqFile in read_file:
    for name, seq, qual in mp.fastx_read(fastqFile): # read a fasta/q sequence
        totalReads += 1
        totalBases += len(seq)
        ...
```

Evaluating the Minimap2 Output

With the mappy wrapper for Minimap2 it is now easy to count the features of interest directly from the corresponding sam file for each reference:

```
for name, seq, qual in mp.fastx_read(fastqFile): # read a fasta/q sequence

hasHit = False

totalReads += 1
totalBases += len(seq)
readLengths.append(len(seq))

for hit in a.map(seq): # traverse alignments
    hasHit = True

    alignmentBases += hit.ctg_len
    alignedBases += hit.mlen

    alignedLength += hit.blen
    ...
```

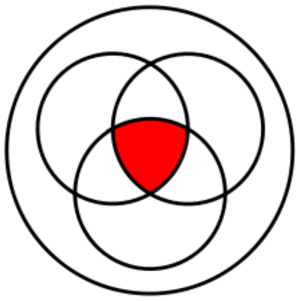
ContamTool.py Output

The read file is assest for each reference. ContamTool.py produces three images per reference from the generated data. A read length distribution of the original FastQ file/ files and two pie charts showing the percentage of aligned and not aligned reads or bases. The collected data, as well as the paths to the images are dumped in a JSON file for easy handling in the further steps.

```
{
"/pathToReference/ecoli_k12_mg1655.fasta":
    {
        "totalReads": 7,
        "alignedReads": 0,
        "totalBases": 62387,
        "alignmentBases": 0,
        "alignedLength": 0,
        "idAlignedReads": [],
        "idNotAlignedReads": ["c9a72623-c55c-4464-ac5e-d1e70cea8466", "4b57cb5c-0c3d-
↪4650-9d57-c94cf4aea2ef", ...],
        "readLengthPlot": "/outputPath/file2_ecoli_k12_mg1655_ref1_ref2_reads_length.
↪png",
        "readsPie": "/outputPath/file2_ecoli_k12_mg1655_ref1_ref2_read_pie.png",
        "basesPie": "/outputPath/file2_ecoli_k12_mg1655_ref1_ref2_bases_pie.png",
        "refs": ["/pathToReference/ecoli_k12_mg1655.fasta"]},

"/pathToReference/ref1.fasta":
    {
        "totalReads": 7,
        "alignedReads": 0,
        ...},

"/pathToReference/ref2.fasta":
    {
        "totalReads": 7,
        "alignedReads": 0,
        ...}
```

Extracting Read Files

Besides the contamination evaluation, *sequ-into* furthermore allows for a separation of the reads into the ones that aligned to the reference versus the ones that that did not align. It generates new FastQ files according to the users inquiry which can then be used in a more elaborate downstream analysis. One notable possibility that *sequ-into* offers, is the extraction of reads against several references at once. Exporting only those reads in the end that represent the intersection (red) of reads aligned against all references or none, according to set theory.



CHAPTER 6

Licence