**MICAS v0.01 User Manuel**

by Tayab Soomro and Matthew Links, with support from Samuel Horovatin

**What is MICAS?**

MICAS was created to aid those wishing to set sequence specific alerts on a nanopore sequencing run. Leveraging a python backend, react.js frontend, and minimap2 for aligning of the user specified sequences, MICAS alters a user specified email when a threshold ( *# of residues matched / residue length* ) is met. MICAS is capable of both runing remotely, on a server or desktop, or running natively on a compatible nanopore sequence (currently only the [Oxford Nanopore Minon Mk1c](https://nanoporetech.com/products/minion) sequencer has been supported).

**What is this Document?**

This document serves as a visual overview of the MICAS system, with an example workflow. This document assumes the end user has a basic understanding of how to work and execute commands in a unix based work environment. For future updates to either MICAS or this document, please check the projects GitHub (<https://github.com/coadunate/MICAS>).

**Installation**

**Prerequisites:**

* macOS or Ubuntu (Unix based is key)
* [git](https://git-scm.com/)
* [conda](https://docs.conda.io/en/latest/)
* A web-browser (any modern browser will do)

**If you are installing MICAS on an Mk1c**, there are a couple preliminary steps required to get the system ready to run MICAS. Please follow the guide posted [here](https://community.nanoporetech.com/posts/getting-conda-running-on-a) (requires ONT community account to read) or the guide within appendix 1 before continuing.

1. Download project from github (<https://github.com/coadunate/MICAS>)
2. Once downloaded, go into the project and run./install.sh. This install script will handle the system step, including the creation and package population of a conda environment.

At this point, installation is more or less complete. If for any reason the **install script failed**, you may **optionally** run the following commands from the projects base folder:

1. Create micas Conda environment:

conda env create -n micas\_env python=3.9 -q -f ./server/environment.yml

1. Enter Conda environment:

conda activate micas\_env

1. Install node.js dependencies:

npm install --prefix ./frontend ./frontend

1. Build node.js dependencies:

npm run build --prefix ./frontend

**Running MICAS:**

Once installation is complete, follow the below steps in order to get MICAS running (NOTE, open each step in a separate terminal):

1. Start redis server:

conda activate micas\_env

redis-server

1. Start celery:

conda activate micas\_env

cd ./server/app/main/utils

celery -A tasks worker --loglevel=INFO

1. Start frontend:

conda activate micas\_env

cd ./frontend

npm run start

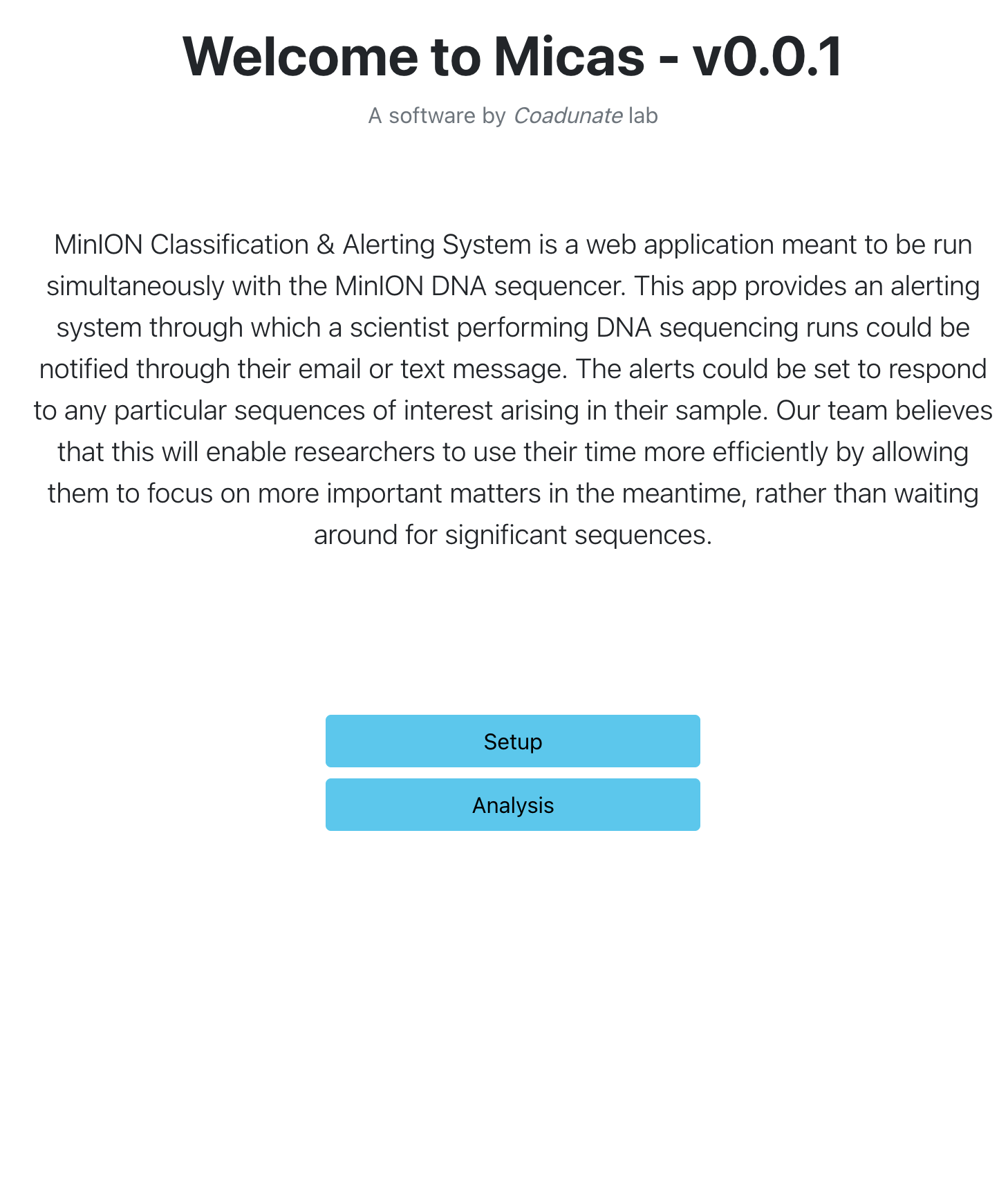
1. Start python server:

conda activate micas\_env

python server/micas.py

At this point, if all went as expected, you should be able to view MICAS’s splash page at <http://127.0.0.1:5000>.

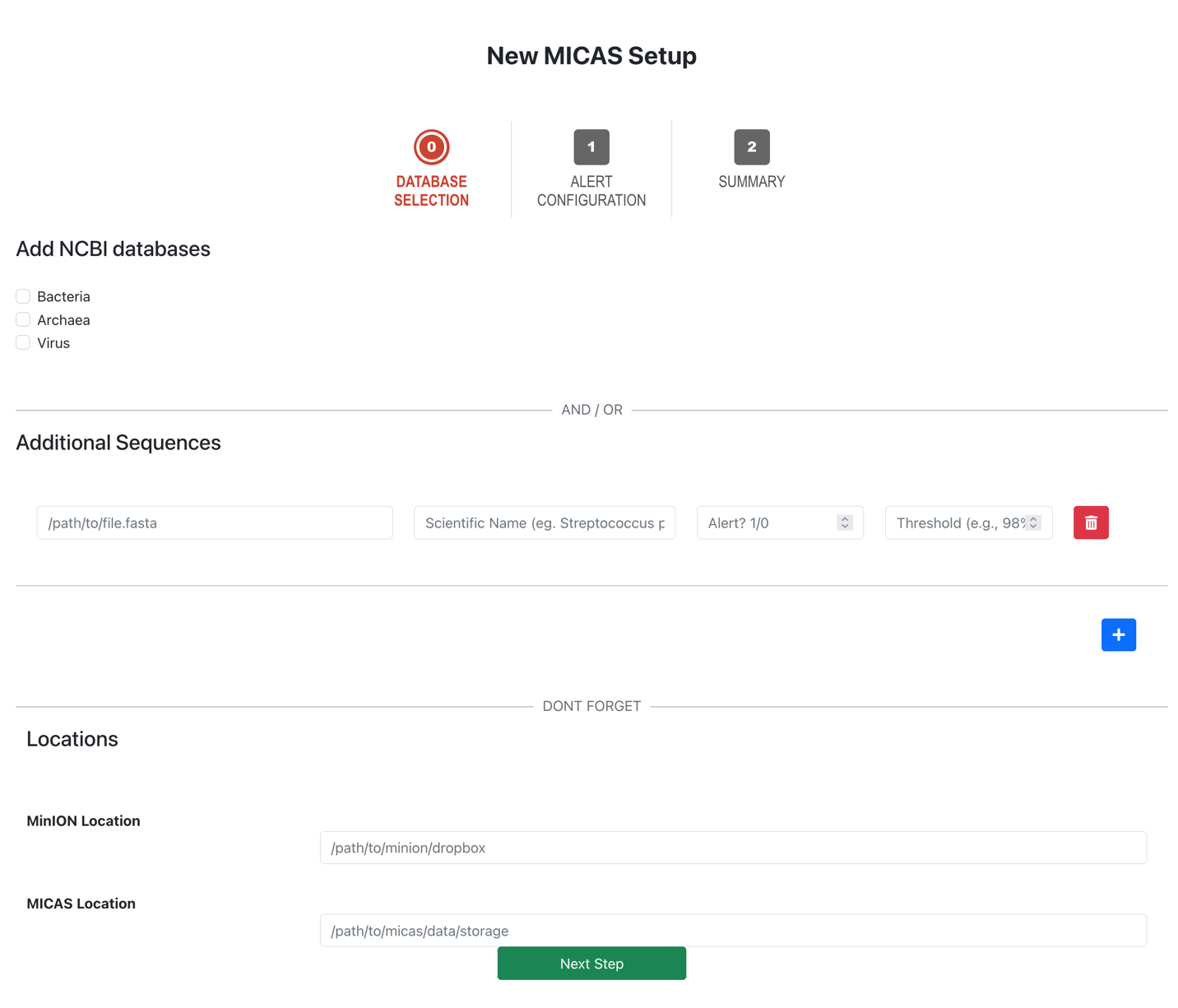
**Splash Page:**



*Link to Setup page. This is where you can setup a new MICAS alert. If this is the first-time using MICAS, start here.*

*Link to Analysis page. This is where you can view currently running or historic MICAS runs.*

**Setup Page:**



*Location of the to where MICAS will store its data files.*

*Location of the to where fasta or fastq files will be dumped post basecalling.*

*Add new sequence report line*

*Delete sequence report line*

*Alert flag. 1 indicates to alert end user, 0 to track up remain silent*

*Name of taxa to appear in analysis report*

*Path to reference fasta file for use in minimap2*

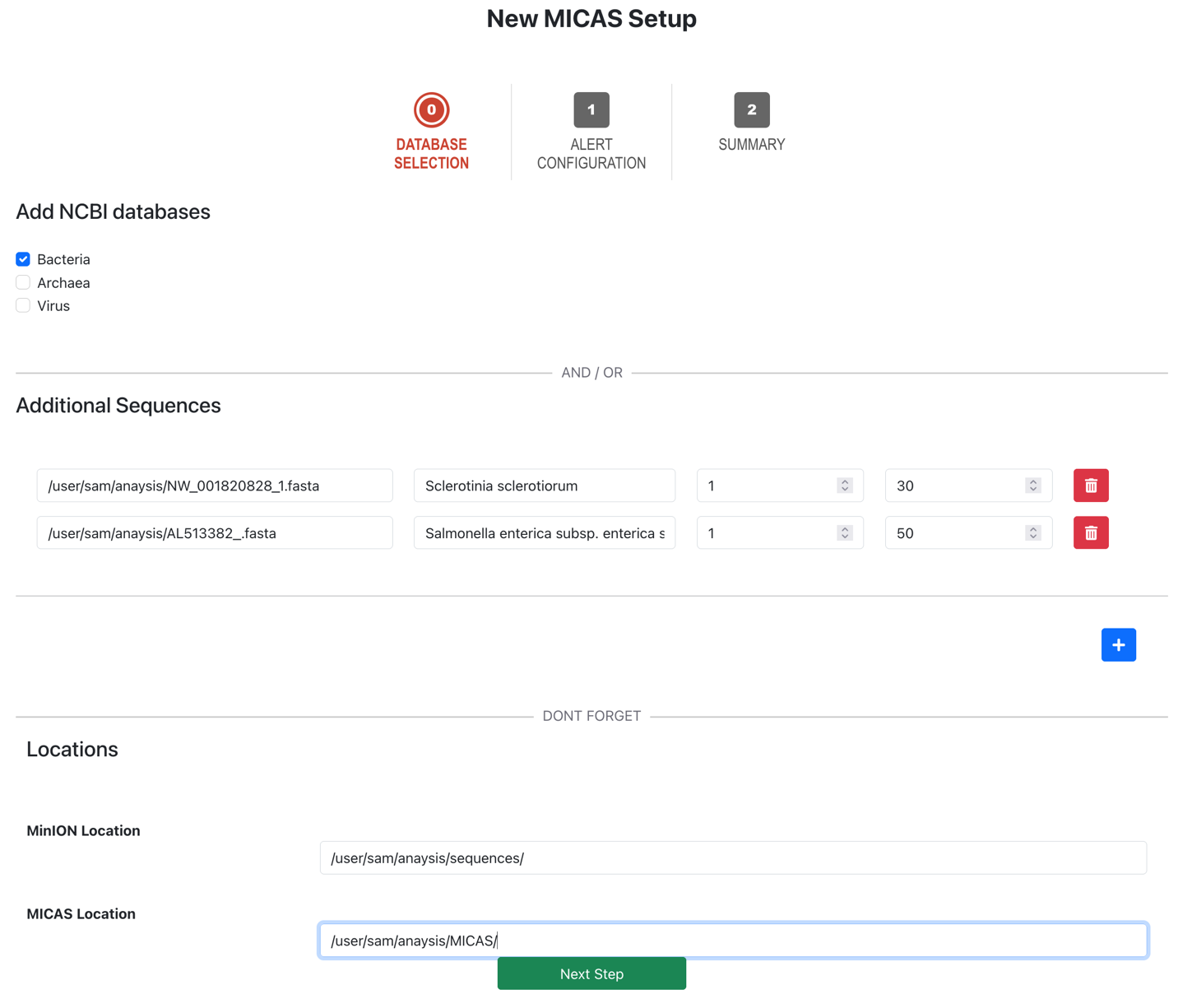
*Have MICAS download a taxonomy database from NCBI for use in characterizing sequenced community. Useful in metagenomic analysis*

*Button to progress to alert configuration page.*

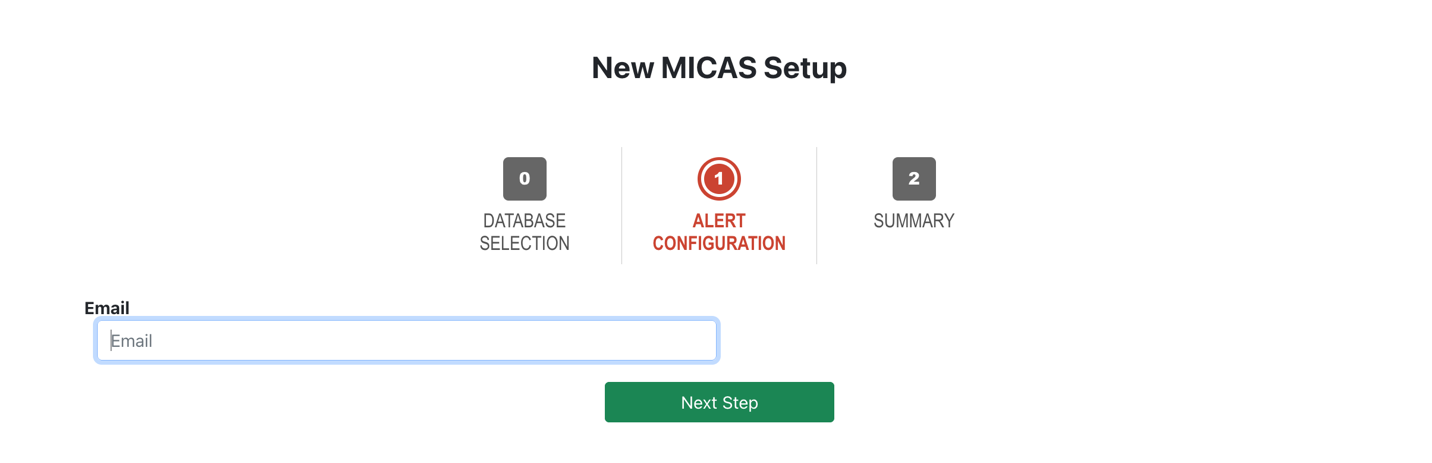
This is the setup page, containing all the user defined search metrics. Given the fasta files, an alert flag, and a threshold ( *# of residues matched / residue length* ), MICAS will monitor the given “MiniON Location” for new or updated fasta or fastq files. Once one is found, MICAS utilizes minimap2 to align the given sequences, and alert the user upon reaching a respective threshold.

If selected, the “Add NCBI databases” will add relevant taxonomic information of the sequenced community in the subsequent analysis.

**Setup Page (Example):**

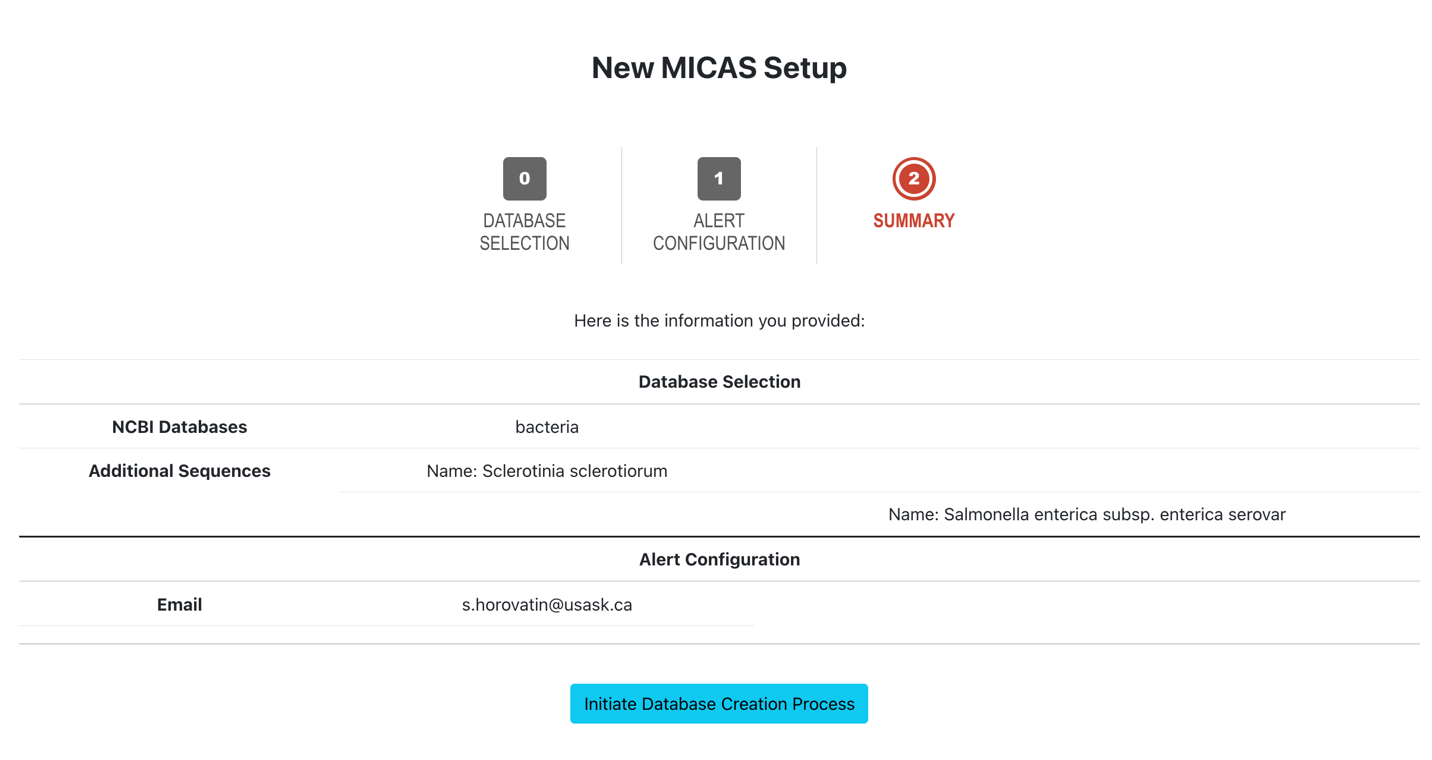


**Notification Page:**



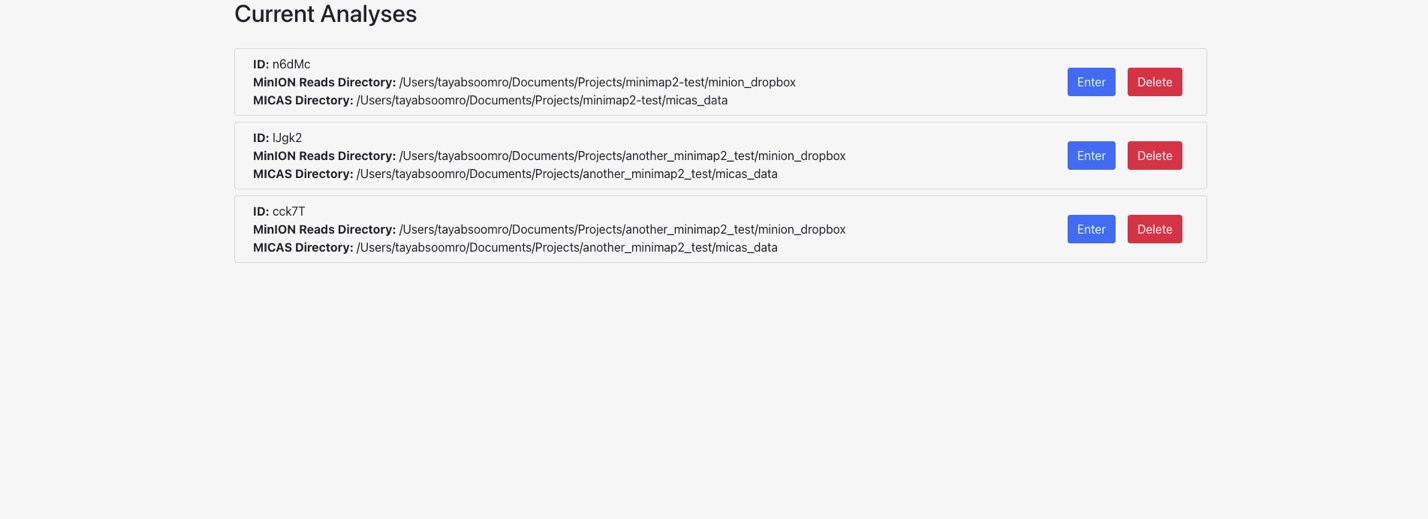
This is the notification page for MICAS, where users can enter the recipient email for any notifications setup within the previous “Database Selection” page. Select next step when you are ready to review the submitted MICAS run.

**Summary Page:**



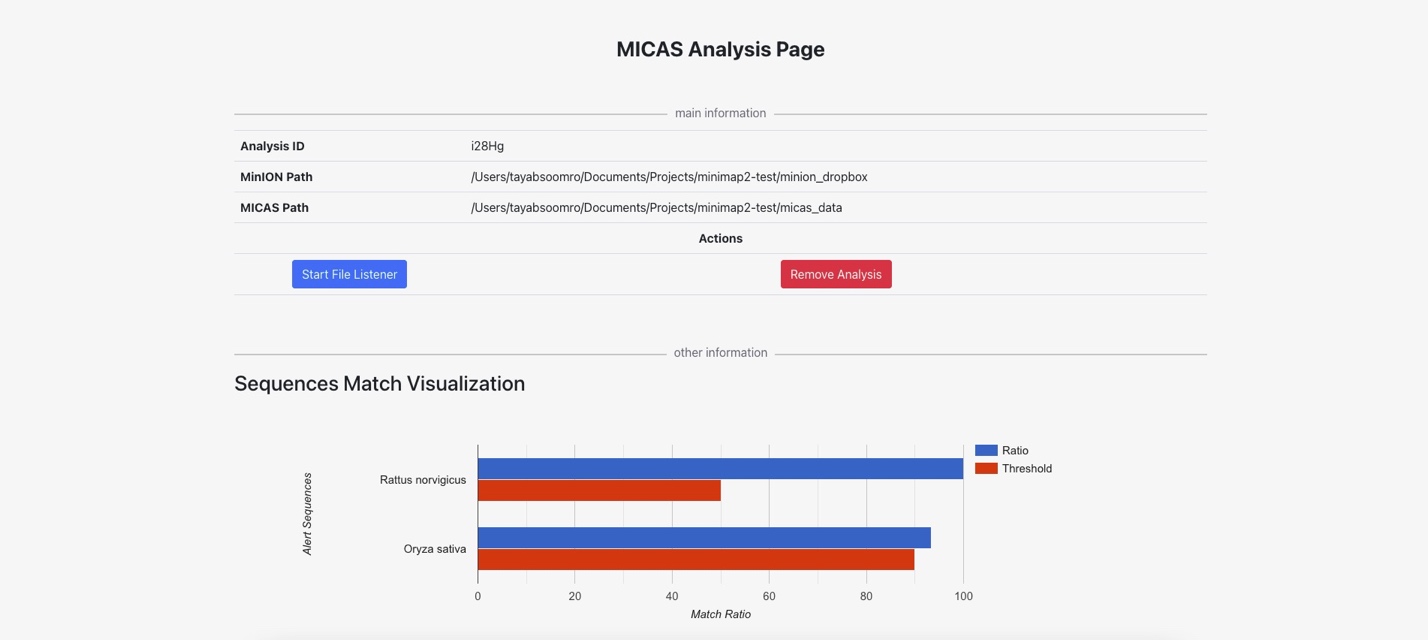
Above is the summary page, reviewing all information input into MICAS in the previous to setup stages. To initiate the creation of the reference database for use by minimap2 based off the provided sequences and NCBI database, select the “Initiate Database Creation Process” button. This also initiates the alert system for MICAS.

**Current Analysis Page:**



This is the current analysis summary page, where all the currently running or historically run analysis can be seen (prior to being deleted by the user). General information, such as automatically assigned ID and directory information are included in each analysis enter. The user can also select to drill down into a specific analysis for more information using the “Enter” button, or delete the entry and terminate the run via the “Delete” button.

**Analysis Summary Page:**



This is a analysis page for a dummy MICAS run. Displayed on this page is general information, such as automatically assigned ID and directory information, a sequence match visualization, and options to start or remove the analysis. To start the file listener and begin matching sequences, select the “Start File Listener” button. To remove the analysis and all associated records from MICAS, select “Remove Analysis” button. Displayed at the bottom of the page is the match ratio visualization, displaying all sequence alerts specified, there thresholds, and the number of sequences presently matched to the alert sequence.

**Standard Example Flow Through MICAS:**

1. (Pre-experiment) Collect relevant reference sequences and calculate appropriate match alert thresholds.
2. Start MICAS given the above instructions.
3. Navigate to the first setup stage, “Database Selection”.
4. (Optional) Select NCBI databases where appropriate.
5. Provide paths, scientific names, alert flags, and thresholds to each sequence of interest.
6. Provide the location of the sequences being generated or existing sequences to be matched against.
7. Provide the location to save all relevant MICAS files.
8. Proceed to “Alert Configuration” and provide an email to be alerted to.
9. Proceed to summary page, review data, and “Initiate Database Creation Process”.
10. Navigate to the “Analysis” page (either through the main page or directly after initiating database creation) and select the newly created run.
11. Select “Start File Listener”.
12. Watch provided email for alerts from MICAS.

**Appendix 1:**

**Getting The Hardware Environment Ready:**

As of writing this guide, the Mk1c has an incompatible version of *glibc*, which is an implementation of the C standard libraries, and this causes the instillation script of Conda to fail. Worse still is that the version of *glibc*present on the system is tightly coupled to many OS functions. Changing the version could cause many fundamental Linux commands to fail if updated arbitrarily. Thus a straightforward and safe method I have found for installing a newer version of *glibc* (without messing with the base operating system version of *glibc*) is to set up a '*debootstrap*' sub Linux system with a newer version of Ubuntu than that which is installed on the Mk1c. If you are unfamiliar with the process of setting up a Debian based OS within a sub-directory of another system, I recommend doing some research before continuing. All the below was done within the */data*folder on the Mk1c.

1. sudo apt update
2. sudo apt install --assume-yes debootstrap
3. mkdir chroot-ubuntu
4. (note, for this example, I have selected the latest Focal Fossa, however you can select any version that best suites you needs and includes a version of *glibc >= 2.25*)

sudo debootstrap --variant=buildd focal chroot-ubuntu

1. sudo mount -t proc /proc chroot-ubuntu/proc sudo mount --rbind /sys chroot-ubuntu/sys
2. sudo mount --rbind /dev chroot-ubuntu/dev
3. **Activate Chroot**(you will want to do this every time you wish to run conda)

sudo chroot chroot-ubuntu /bin/bash

At this point you will be ready to move on to getting Conda set up, however **if you wish to remove the above setup**and start over, you can use the following:

1. sudo umount chroot-ubuntu/proc chroot-ubuntu/sys chroot-ubuntu/dev
2. sudo rm -rf chroot-folder

**Setting Up Conda In Chroot Home:**

At this point, we will have successfully set up a Ubuntu instillation within a sub-directory of our Mk1c. All changes we make the system's configuration from now on will be limited to this new Ubuntu sub-instillation.

With the *glibc*issue resolved (as the Focal Fossarelease of Ubuntu includes an up to-date version of*glibc*)we may proceed with the Conda installation.

1. cd home
2. apt install curl
3. (note, the Conda version in this tutorial must be 4.9.2, as more recent versions will instillation issues)

conda curl -o miniconda37.sh https://repo.anaconda.com/miniconda/Miniconda3-py37\_4.9.2-Linux-aarch64.sh

1. bash ./miniconda37.sh
2. Install Conda via installation script prompts
3. To test install, run *conda info.* If Conda succeeds at providing you information on the newly installed Conda implementation, you may move on to the next steps.

At this point you have successfully installed Conda on your Mk1c: well done! Proceed with application/pipeline specific installation and environment setup. The next steps are for those who wish to install Python within their Conda environment:

**Set Up Python In Conda:**

1. (note, any version of Python can be specified here. I have tested with versions 3.5-3.9, with all functioning as expected)

conda create -n pyenv python=3.7

1. conda activate pyenv
2. (note, Conda fails to recognize the required dependency for openssl, which causes Conda to fail to run python. This can be remedied by specifying a functioning version of openssl)

conda install -y openssl=1.1.1j=hfd63f10\_0

At this point, should now have a functioning installation of Conda, with an environment running whichever version of Python you wish.