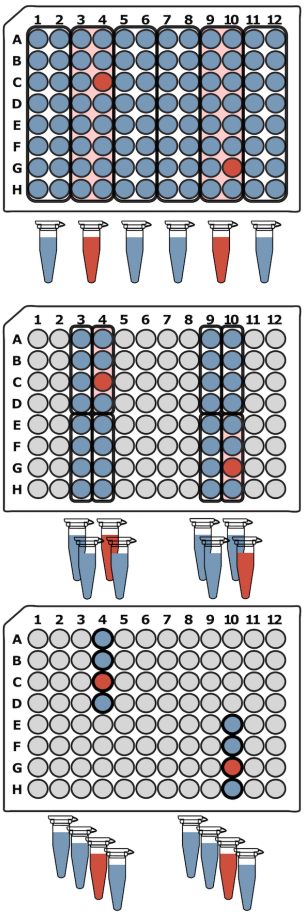
CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: Sample Pooling Protocol Application to manage pathogen testing	
Sponsor Information:	Dr. Tara Furstenau, Assistant Research Professor Tara.Furstenau@nau.edu Dr. Viacheslav Fofanov, Associate Professor Viacheslav.Fofanov@nau.edu School of Informatics, Computing, and Cyber Systems Northern Arizona University

Project Overview:



Despite the rapid development of tests that detect the presence of SARS-CoV-2, the ongoing national testing rollout has revealed several major challenges of large-scale pathogen surveillance efforts. The rapid spike in nationwide testing led to reagent and equipment shortages which placed severe bottlenecks on testing throughput. Initially, this meant that testing priority was given to samples that were highly suspected of being positive. Now, as we begin to open up and return to work and school, it is essential that we expand testing capacity to capture a larger population so that we can better track, isolate, and contain outbreaks. However, performing routine testing on large numbers of asymptomatic individuals where a very large percentage of them are likely to be negative feels wasteful given the current limited supply of reagents and other materials. One promising technique to make large-scale screening more efficient and economical that is being explored by many labs is based on *group testing theory* (i.e. testing pooled groups of samples rather than testing them each one-by-one).

Group testing was first proposed in WWII as a way to screen out syphilitic men from military service. After seeing that the US was performing individual blood tests on millions of specimens in order to detect just a few thousand cases, Robert Dorfman proposed that more information could be gained per test if many samples were pooled together and tested as a group. If the test performed on the pooled sample was negative (which was very likely), then all of the individuals in the group could be cleared using a single test. If the pooled sample was positive, the samples would need to be re-tested individually.

Since then, group testing theory has advanced and many different approaches have been developed with varying strengths and weaknesses. Typically, in an effort to minimize the number of tests required, some approaches have become too complex to be accurately carried out without the aid of robotic equipment, while other approaches require too many steps that take too much time. We have recently developed our own protocol which is still highly efficient in the number of tests but also focuses on being relatively easy to perform by laboratory technicians in just a few steps. In the modified 3-Stage pooling example (see insert), we begin by creating 6 pools with 16 samples each; the positive pools from the first step (red dots) are then subdivided into 4 groups of 4 in the second step; in the final step, the samples from the positive pools in step 2 are tested individually. The total number of tests, as well as the size of the pools depends on the total number of samples, positivity rates, and the arrangement of the positive samples.

Because our focus is on making our pooled testing technique as easy and error-proof as possible for the overtaxed laboratory workers, we would like to augment it with an easy-to-use GUI-based tool to capture/guide their work, as well as documenting correctness and helping to efficiently capture the results. In this sense, the desired product is a *workflow assistant* that supports lab technicians at all steps of the process. Ideally, we would like to have this assistance begin right at the protocol design phase, i.e., in assisting the researcher in designing an optimal pooling

strategy based off of our existing algorithm and then embodying this protocol in the in-lab tool to keep track of samples as the protocol is executed.

Specifically, we envision a cross-platform mobile application (Android or iOS) that can run on either a smartphone or a tablet carried by the researcher or lab technician. Initial prototypes can be for Android to ease development logistics, but team is encouraged to use a cross-platform framework like Ionic or React Native to allow easy generation of functional iOS version from same codebase.

Some core functionalities for a minimum usable product would include:

- Portability, the researcher should be able to use the app on a smart phone, tablet or laptop computer.
- Allow user to quickly design and visually display the optimal protocol (similar to the inset figure).
- Visual display of the configured protocol, as well as user's progress through it.
- Provide a touch interface for the user to mark samples that have been processed as they work.
- Allow the user to update the protocol based on the outcome of each test.
- Ability to save progress. The protocol takes place in three stages so the user should be able to revisit a saved protocol and continue where they left off.

Additional features that would make it truly usable:

- More refined GUI for visualizing protocols, monitoring protocol execution, and reviewing results.
- Include secure user accounts so that user accounts, including a user profile system where users can set default preferences for many parameters of their particular studies.
- System provides multi-function "lab notebook" for researchers. Users can save/edit/access many aspects of their work, e.g.:
    - Protocols that they have previously designed can be saved/cloned/edited and re-used.
    - Can save results from different experimental runs
- Add additional user-provided constraints and modify the protocol algorithm to accommodate the constraints.
- Provides a "group space" where lab users can post/share protocols or results with others in the lab.
- iOS version of app generated, verified, and tested.

This application will help facilitate efficient and accurate sample pooling for high throughput processing of pathogen samples. The impact of this product can easily extent past the current COVID-19 pandemic and be used for many pathogen surveillance campaigns, particularly in resource-poor settings (e.g. pathogen surveillance in developing countries, or in non-human systems, such as wildlife disease surveillance).

Knowledge, skills, and expertise required for this project:

- Programming and software development skills for modern web interfaces and databases.
- Development of graphical user interfaces in web-based systems, including end-user testing/refinement.

Equipment Requirements:

- There should be no equipment or software required other than a development platform and software/tools freely available online.
- Client will provide basic algorithm for designing the pooling protocol.

Software and other Deliverables:

- A complete software product, fully tested.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- Users guide to running the code, appropriate for someone with little software background.
- Complete professionally-documented codebase, delivered both as a repository in GitHub, BitBucket, or some other version control repository; and as a physical archive on a USB drive.