EpiMobile: Pathogen Point of Care Diagnosis and Global Surveillance using Mobile Devices

Rodrigo Araújo
rodarauj@cs.ubc.ca

Alexandra Kim
kimalser@cs.ubc.ca

Anamaria Crisan
acrisan@cs.ubc.ca

ABSTRACT

Rapid identification of disease causing infectious agents (pathogens) in patient samples is important to mount effective responses to potential, or ongoing, disease outbreaks. New advances in genomic sequencing technologies are enabling more effective point-of-care diagnosis – where patient samples can be analyzed for the detection of pathogen DNA. However, it has yet to be understood how these technologies can be applied *in the field* and under resource constraints that hinder cloud access for computation. Here we present EpiMobile, a conceptual model and minimal viable product (MVP) implementation of a genomics point-of-care workflow using mobile devices. EpiMobile enables analyses of genomic data harvested by a portable genome sequencer and the distribution of analysis results to local clinical or healthcare teams as well as national, or global, public health agencies, whilst considering computational processing and Internet connectivity resource constraints. Our evaluation of EpiMobile indicates that it has a minimal resource consumption footprint and is accurate when run of a dataset with known outcomes. However, we emphasize that the conceptual and exploratory nature of our work affects to what extent our results would map to real world settings. We discuss the utility of EpiMobile through a set of usage scenarios currently supported by our MVP implementation. We believe that our work provides an interesting overview of an exciting and emerging healthcare application context as well as proposing an interesting implementation of genomics point-of-care.

CCS Concepts

•Computer systems organization → Special purpose systems; •Applied computing → Health care information systems;

Keywords

Mobile diagnosis system; system design; disease prediction; outbreak control.

UBC CPSC 538w-2017, Data at Scale, Vancouver, BC CANADA

ACM ISBN 123-4567-24-538/04/26…$1,000,000,000

DOI: 0.0.0.127

1. INTRODUCTION

Genomic data has the potential to improve patient diagnosis and public health efforts if this data can be obtained and rapidly analyzed at the time when a clinician is assessing a patient (the point-of-care). However, genomics point-of-care is not currently a reality. Contemporary public health systems rely on the output of laboratory tests to confirm the diagnosis of a communicable disease, but these tests have slow turn-around times that introduces diagnosis delays [19, 33]. Some public health agencies are exploring the use of high-throughput genome sequencing technologies, which provide a more rapid turn around with an accuracy that is comparable, and sometimes even better, than existing laboratory tests [14, 31]. However, even these genomic sequencing technologies remain confined to the laboratory, due to their size and resource demands, and thus are difficult to apply towards point-of-care applications. It is only very recently that portable genome sequencing technologies have been developed. Early publications using the MinION portable sequencer *in the field* during the 2014-2016 ebola outbreak [32] and the (currently) ongoing Brazillian zika outbreak [17] have demonstrated the potential of this technology but have also revealed a number of challenges, especially unanticipated resources challenges, that remain to be solved. Looking further into the future, it will be necessary to establish how mobile devices, such as smartphones, combined with portable sequencing technology could be effectively leveraged to enable true genomics point-of-care in a variety of resource availability settings. These resource constraints can be in the form of computational power, to Internet connectivity, to battery power, and while these constraints are well established by the systems research community [34] they are relatively absent considerations from the bioinformatics community that drives genomic research and relies on resource intensive compute systems.

Here we present EpiMobile, a conceptual sketch with a minimal viable product implementation of a mobile device workflow for genomic point-of-care diagnosis of communicable diseases. We limit the scope of our application context to diagnostic tasks and simple sharing of analysis results with local healthcare teams and remote servers of national or global public health systems. Our implementation *speculates* on the diagnostic potential of portable genomic sequencing devices, namely MinION [24, 5], which weighs only 100g and its (still conceptual) cousin SmidgION, which is designed to operate on smartphones [7]. For our research to be applicable to real world genomics point-of-care contexts, EpiMobile must be able to adapt to varying resource constraints in

order to share data effectively with local healthcare teams and national, or global, public health agencies. Resources requirements will vary greatly depending on the particular usage scenario, and a truly generalizable system must be able to work in the field out in remote geographic regions as well as in highly resourced hospital settings.

Our work provides two major contributions to the existing distributed systems and bioinformatics literature:

- Description of an end-to-end system that identifies pathogens from a patient's sample during a point-of-care interaction and shares these results with local healthcare teams and central public health agencies, and
- Exploration of cross-layer interactions between data processing, analysis, and reporting layers in light of computational power and Internet connectivity resource constraints.

The rest of this paper is organized as follows. Section 2 describes the public health application context for our work and provides a high-level overview of genomics and bioinformatics. Section 3 illustrates the conceptual overview of our system, with specific implementation details in Section 4. We conduct benchmarking experiments in Section 5 to assess resources consumption by our system, and also to demonstrate its diagnostic performance on dataset for which we have known the outcomes. Note that in the evaluation we were not able to run our system on a actual mobile device because some components necessary to run our application on a mobile device are not currently complete but will be in the near future (details in Section 4), thus our discussion of evaluation results is couched in some speculation of how our system would perform on a mobile platform. Assumptions we made for our conceptual sketches and implementations are in Section 6. In Section 7 we discuss the utility of our system through a set of usage scenarios concerning pathogen identification and local vs global pathogen surveillance. We present related work in Section 8 covering current field deployed mobile pathogen point-of-care genomics and surveillance implementations and also other research from biological and non-biological sensors networks deployed on mobile phones or other devices. Finally, we summarize our project and outline future work for EpiMobile system in section 9.

2. BACKGROUND

In this section we provide an overview of our application context, genome sequencing, and bioinformatic analysis. We assume that the reader may have only passing knowledge of these subjects, more knowledgeable readers can therefore safely skip this section (or read it anyway and silently pass judgment on how we've distilled these concepts for our audience).

2.1 Public Health and Clinical Medicine

Healthcare system comprises two domains - clinical medicine and public health - that have separate aims, but must work together to improve the health of individuals and whole populations [1]. Clinical medicine is siloed and consists of specialized practitioners (doctors, nurses, pharmacists, technicians) that diagnose and treat a single individual. Public health, in contrast, performs disease prevention and control activities that target entire populations, and has a very diverse group of practitioners (community leaders, politicians, clinical medicine practitioners). For example, in an outbreak of a communicable disease, like ebola, public health agencies are responsible for containing the spread of disease and preventing new cases (individuals that become infected), while clinical medicine practitioners treat sick patients. In practice, the boundaries between the two domains are fluid, but being aware of these differences and distinguishing between them is important to appreciate why technological tools for public health may not be appropriate or sufficient for clinical medicine, and vice-versa.

Our work emphasizes a public health application context, and more specifically disease diagnosis (relevant also to clinical practitioners) and surveillance, the practice of collecting information to monitor the spread of disease, the likelihood of an outbreak, and the potential severity of the outbreak [21].

2.2 Genome Sequencing

Genomic sequencing technology refers to the full complement of procedures, platforms (genome sequencing machines), and analysis tools necessary to convert DNA in biological matter to a digitized form that is then computationally analyzes. While some manner of genomic sequencing has existed for decades, it has only been since the mid-2000s, and notably marked by the completion of the first draft of the human genome, that so-called high-throughput genome sequencing technologies (also referred to as Next Generation Sequencing (NGS)) became available, affordable, and much faster than their contemporary alternatives [20]. Over the course of nearly two decades, high-throughput genome sequencing has begun to supplement traditional laboratory technologies (for example [31, 18]). This transition toward high-throughput technologies has made it possible to rapidly analyze the whole genomes of humans and pathogens (disease causing microorganisms), and has enabled exciting possibilities to change public health and clinical practices [13], including genomics point-of-care applications.However, the full transition toward high-throughput genomic technologies is not yet complete, and considerable research is still being carried out to validate this technology and assess its limits.

2.2.1 MinION and SmidgION

There exist a number of genome sequencing platforms, each with their own strengths and limitations. Our work focuses on genome sequencing with a new platform called MinION [24, 5], and in fact through our work we attempt to anticipate the utility of a yet-to-be released MinION cousin, SmigdION [7], which is a genome sequencer built to operate with mobile phones as opposed to relying on desktop or laptop computers. The primary differentiators between MinION (and by extension SmigdION) is the nanopore technology that, among other things, has enabled the development of a truly portable genome sequencer; MinION can fit in the palm of an adult's hand, whereas other genome sequencers are considerably larger and can only be operated in specialized laboratories. This allows MinION to be a very versatile genomics point-of-care device, as was demonstrated in two field studies using MinION during the West African ebola [32] and Brazilian zika [17] outbreaks.

Our research seeks to explore how a genomics point-of-care workflow (Figure 1) that encompasses nanopore tech-

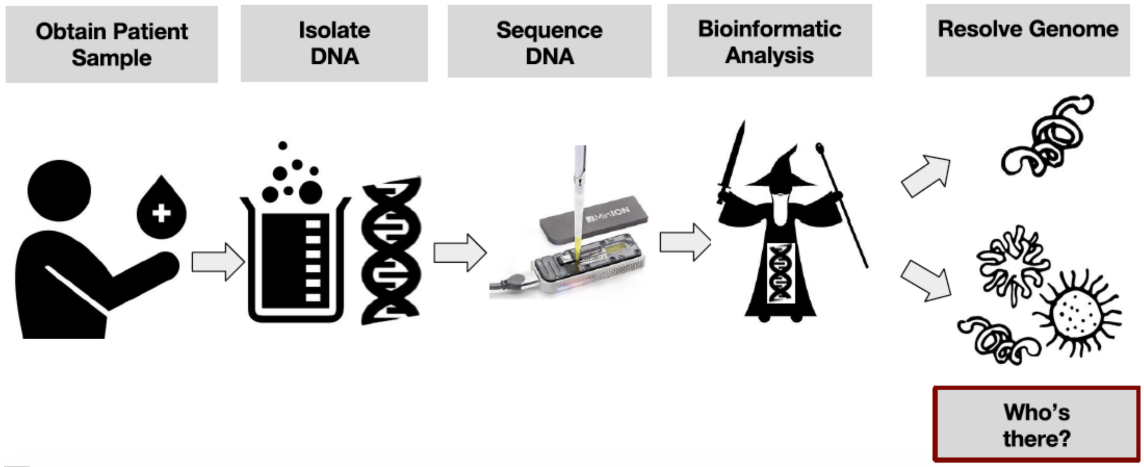

Figure 1 An overview of genomics point-of-care workflow. At a high-level there are five high-level steps to conduct a genomics point-of-care assessment. The first two steps are preparatory : obtaining a sample (blood, urine, saliva, etc.) from a patient, and then *preparing* the sample for analysis by isolating DNA content from other biomolecules in the sample (i.e. proteins). Following sample preparation is the sequencing step, where the organic biological material becomes a digitized string of DNA comprising A,C,T,G letters (referring the nucleic acid building blocks of DNA). Note that other types of genomic biomolecules (RNA) can also be sequenced, but we do not discuss those here. Finally, once the DNA is digitized there is a bioinformatic analysis step that can yield one of two outcomes: a genomic sequence for a single pathogen, an assessment of all the organisms in the sample (metagenomics, see Section 2.3.1). Our systems design is focused on *"who's there?"* metagenomics approach. *Figure icons are courtesy of the noun project.*

nology could be run on a mobile device and how results can be shared beyond the device that conducted the analysis. Our work *does not* comment on the procedures required to prepare a biological sample (Figure 1) to be analyzed by the genome sequencer[1]. While quality of data from these portable sequencing technologies do not yet match their much larger, laboratory confined, counterparts, advances in the underlying nanopore technologies continue to refine the data quality [23] to make MinION an important technology for genomics point-of-care.

2.3 Bioinformatics

The discipline of bioinformatics emerged with the demand to analyze, interpret, and manage the large amounts of genomic data that was being produced by high-throughput genome sequencing platforms. Cross-cutting biology, computer science, and applied statistics, bioinformatics is a diverse and rapidly changing discipline. Although there exists commercially available bioinformatic software, the majority is open source of varying degrees of quality and levels of maintenance.

2.3.1 Metagenomics

Metagenomics broadly refers to establishing an inventory or micro-organisms present in an environmental sample based upon the nucleic acid content recovered from the sample [37]; in very simplistic terms metagenomics allows researchers to assess *"who's in there"* based upon whatever genomic material is detectable in the sample. Metagenomics is in contrast to more traditional methods that target a specific microorganism, usually through culture techniques (growing microorganism in lab from patient sample), in an attempt to reconstruct its genome for further analysis. Although theoretically possible for a long time, metagenomics is a relatively new area of study within genomics and bioinformatic research largely due to the challenges of developing computational methods to classify recoverable genomic content to some specific organism and the need for reference databases of organism genomes [38]. It is anticipated that in the future of medical and public health testing will converge towards metagenomics, foregoing the so-called culture-based methods of contemporary laboratory testing.

Our application context uses a metagenomic approach to attempt to identify whether some pathogen could be detected from a human sample (blood, urine, saliva, etc.). For example, could it be possible to identify ebola in the blood sample of a patient suspected to have the disease but who may not yet be showing symptoms?

[1]This step is typically called library or sample preparation. We have reason to believe that Oxford Nanopore (the company that makes MinION and SmidgION) will eventually provide an easy to use prep-kit. Indeed that is what their VolTRAX technology (`https://nanoporetech.com/products/voltrax`) is attempting to do.

3. DESCRIPTION OF THE SYSTEM

In this Section we describe the design decisions underlying EpiMobile as abstractions and data structures. To facilitate our description of the end-to-end aspects of our system we break down EpiMobile's components into four layers of abstractions: the device input layer, data processing layer, application layer, and communication layer. We also describe the cross layer interactions within EpiMobile. The concrete implementation of these abstractions is presented in Section 4.

3.1 Device Input Layer

The device input layer consists of the physical connection

between the sequencing device and mobile device, and the transmission of data between them. There are many aspects of this layer that are out of our control as it is defined largely by the Oxford Nanopore corporation, and as such do not further describe this layer and refer the reader to Loose *et al.* [25].

3.2 Data Processing Layer

The data processing layer takes as input a data file from the device input layer to be processed by bioinformatic tools. It is possible to accept a stream of input data from the device layer, however, we do not yet provide this capacity within EpiMobile. This layer will analyze the genomic data and attempt to identify organisms that are present in the patient's sample by comparing against a reference database stored on the device. Two sets of tabular analytic results are prepared by this layer. The first is a table of results that reports the full complement of organisms (*organism table*) that were detected in the analysis sample, which will stored in a database and accessed by the application layer. The second is a table of results reports only the top-hit (best match) organism, which is stored in a *case table* on the device. The case table records all of the patients a clinician has seen and is used by the communication layer to transmit results locally and globally.

There are other functions could be implemented in the data processing layer that could be fed to the the application layer, such as automatically monitoring for potential outbreaks and generating alerts, suggesting appropriate treatments, or performing clustering or phylogenetic analysis on the device, however those functions are outside the diagnostic scope we constrained ourselves to.

3.3 Application Layer

The application layer is what a user will primarily interact with to begin analysis, view results, and communicate results. To report results the application layer draws from the results of the full complement of organisms and currently only reports the top-hit organism to a user interface. We do not currently manage conflicts in the event that multiple pathogens may be present in a patient, for example, many patients with tuberculosis can also be co-infected with HIV.

3.4 Communication Layer

As previously indicated, EpiMobile must be able to operate in the environments with very limited Internet connectivity. Therefore the system is designed in such a way that for the local "on-spot" operation where Internet connection is not required. Local data exchange is largely reliant on human protocol to meet and synchronize case table data stored on individual devices. Global data exchange, which we define as sharing data with remote servers, is only done when there is Internet connection.

3.4.1 Local vs. Global Communication

Local communication. The EpiMobile applications will store genomic sequences, an organism table, and a case table on a mobile device. Genomic sequence files are too large to pass between, and store redundantly, on multiple mobile devices, and organism tables provide too much information. As such, we only exchange case table results as part of local communication. This is achieved through human protocols as opposed to automated, gossip based protocols that consume resources "listening" for nearby devices. The human protocols rely on clinical teams to meet up and manually synchronize their devices via the user interface in the application layer.

Global communication. Global communication assumes that a EpiMobile can transmit results to a remote server through a reliable Internet connection. The remote server can belong to a national public health agency, like the United States Centre for Disease Control, or a Global agency such as the World Health Organization (WHO). EpiMobile will transmit both genomic sequences and case table results. Recall that genomic data is *not* synced across mobile devices, and as such a device can only transmit data it has processed. Case tables, if synchronized at sites without an Internet connection, reflect up-to-date case counts. This allows national or global agencies to have to levels of resolution on emerging or evolving outbreak situations: a high resolution view via genomic data, and a cruder lower resolution view via case counts.

Outside of putting data onto these remote servers, EpiMobile does not currently anticipate what kind of analyses public health agencies may perform or expect to fetch data from these central servers onto mobile device, although we explore possibilities in our usage scenarios (Section 7) and Future Work (Section 9.1). However, we do know that these data types are either currently used by public health agencies of will be used more often in the future [19].

3.4.2 The communication protocol

The system follows an eventual consistency model of communication, which means that each device might hold a different set of samples and sample results and/or samples and sample results with different states. The following protocol allows that eventually, the remote server will have all samples and samples results:

- Each item in the pair (sample, sample result) after being analyzed locally is marked as *GloballyNotShared*;
- Whenever a device gets Internet connection, it sends its genomic sequence files and case tables that are marked *GloballyNotShared*, and marks them as *GloballyShared*;
- Whenever a device connects to a local peer device, it shares all of its case tables, together with their marks of *GloballyShared* or *GloballyNotShared*. This goes both ways, i.e. the peer also shares its (marked) sample results with this device;
- Whenever a device connects to a local peer device, it updates the peer about their *GloballyShared* case table results, this way this peer will not have to redundantly globally share the sample results that have been globally shared.

The following scenario might better describe the dynamics of the protocol: Suppose we have 3 field researchers with 1 device each, let them be $N1$, $N2$, and $N3$. N1 holds samples $[x_1, x_2]$, $N2$ holds pairs $[x_3, x_4]$, and $N3$ holds pairs $[x_5, x_6]$, all with the respective case table sample results, $[r_{x_1}, r_{x_2}]$, and so on, respectively (Figure 2).

- $N2$ gets Internet and shares its pairs, marking $[x_3, x_4, r_{x_3} r_{x_4}]$ all as *GloballyShared*;

- $N1$, $N2$, and $N3$ meet and connect to each other;
- $N2$ shares r_{x_3} and r_{x_4} with $N1$ and $N3$ and it also tells them that $[r_{x_3}, r_{x_4}]$ have been globally shared;
- $N1$ sends $[r_{x_1}, r_{x_2}]$ to $N2$ and $N3$. $N2$ now holds 2 items that need to be shared: $[r_{x_1}, r_{x_2}]$, $N3$ now holds 6 items that need to be shared: $[x_5, x_6, r_{x_5}, r_{x_6}, r_{x_1}, r_{x_2}]$;
- $N3$ sends $[r_{x_5}, r_{x_6}]$ to $N1$ and $N3$. $N2$ now holds 4 items that need to be shared: $[r_{x_1}, r_{x_2}, r_{x_5}, r_{x_6}]$, $N1$ now holds 6 items that need to be shared: $[x_1, x_2, r_{x_1}, r_{x_2}, r_{x_5}, r_{x_6}]$.
- They disconnect from each other and split up;
- $N3$ gets Internet connection and globally shares $[x_5, x_6, r_{x_5}, r_{x_6}, r_{x_1}, r_{x_2}]$ and marks them as *GloballyShared*;
- They get together again and connect to each other, $N3$ tells $N1$ and $N2$ that the items $[r_{x_5}, r_{x_6}, r_{x_1}, r_{x_2}]$ have been shared. Note, that it is not necessary to inform about the sample genome files (i.e. x_5 and x_6), because those are never shared locally due to their file size (Section 3.4.1).
- Eventually, $N1$ gets Internet connection too and shares the remaining $[x_1, x_2]$.
- Now, the remote server has all the data.

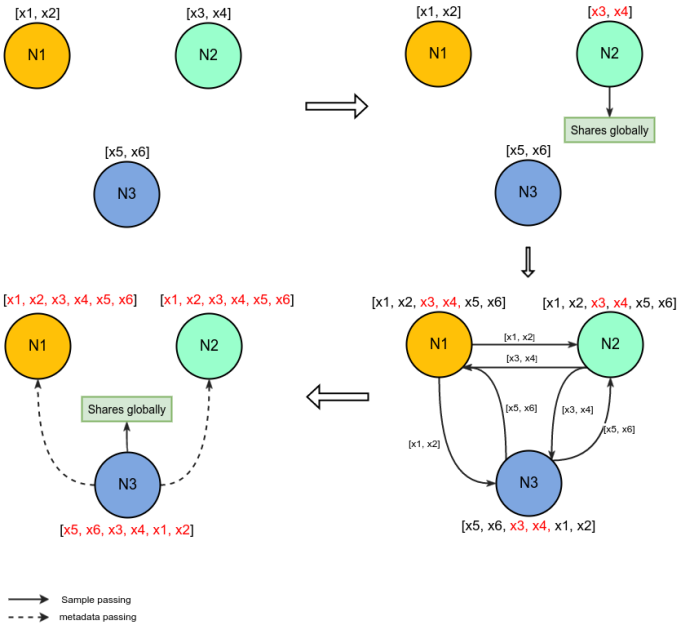

Figure 2 Communication protocol used by nearby devices to share their sample results

3.5 Cross Layer Interactions

Collectively the device input, data processing, application, and communication layers form, in that order, the end-to-end components of EpiMobile from obtaining a patient's sample to communicating the results locally and globally. Data is passed sequentially from the device input to the application layer, however local and global communication relies on a combination of human protocols and Internet availability, respectively.

We intentionally did not explore security or privacy protocols that can also serve to facilitate cross-layers interactions in this implementation. Currently, genomic data is not considered to be patient identification information because this data is freely shared online and is publicly accessible via NCBI (a genomic data repository operated by the United States National Institutes of Health).

4. IMPLEMENTATION

This section describes the tools, methods and ideas used in our implementation of the EpiMobile system.

4.1 Software Container

One of the requirements of the system described in section 1, is to be able to run the system across a multitude of mobile devices. However, it is highly impractical to write a version of our system adapted for each computer or mobile device. For this reason, there is a need for a generalized software container.

Docker [2] is an open-source application container engine that acts as a software container thus providing an additional layer of abstraction and alleviating the need to write different versions of the system, adapting to every device. We are going to use Docker to build a software container to make sure that EpiMobile works across different devices and operating systems.

Docker allows separation of applications from infrastructure so one can deliver software quickly. With Docker, one can package and run a software application in an isolated environment called container. On a given host it is possible run multiple docker containers due its isolation capability. Although in some ways they are similar to virtual machines, containers have a lightweight nature because they run without the extra load of a hypervisor. By using Docker it is possible to simplify the development flow by encapsulating the application and its dependencies into Docker containers; these containers are easy to manipulate and allow the user to reproduce similar behavior on different underlying architectures. The most common uses for Docker are: 1) Consistent delivery of software application; 2) Responsive deployment and scaling; and 3) Running more workload on the same hardware

Docker uses a technology called namespaces to provide the isolated workspace called the container. When you run a container, Docker creates a set of namespaces for that container. These namespaces provide a layer of isolation. Each aspect of a container runs in a separate namespace and its access is limited to that namespace. Docker Engine uses namespaces such as the following on Linux:

- The pid namespace: Process isolation (PID: Process ID);
- The net namespace: Managing network interfaces (NET: Networking);
- The ipc namespace: Managing access to IPC resources (IPC: InterProcess Communication);
- The mnt namespace: Managing filesystem mount points (MNT: Mount);

- The uts namespace: Isolating kernel and version identifiers. (UTS: Unix Timesharing System).

The main reason why we chose Docker is its acceptance by the software industry. Docker has become the most popular container technology, being used by big companies such as Spotify, Yelp, eBay, Expedia, ING, New Relic, The New York Times, Oxford University Press, PayPal, Sage, Shopify, The Washington Post and Uber. We have a positive experience of using Docker, both from the development side and from the user side, since Docker makes running applications easy, even for non-tech savvy people.

4.2 Stack

In our simplistic implementation of the system, we are using Flask [4], a micro web framework for Python. We have chosen Python as our main programming language, largely because among all languages, Python is well known to all three of the authors; it is also a good enough tool for this kind of project, especially because it is straightforward to prototype an application in Python. We are using a PostgreSQL database to store the results table and user interface is shown via a browser using HTML, CSS and JavaScript.

4.3 Communication

Our EpiMobile design anticipates that in a real-world application a stable Internet connection may not be available, and therefore we need to minimize the amount of transferable data and find other ways, a mobile device can communicate with the external world.

Unfortunately, it is impossible to communicate with remote servers without *any* Internet connection however it *is* possible to minimize data transfer to the remote servers as described in section 3.4. We can however synchronize local data with other devices, assuming that those are located within a limited distance range. Synchronization is feasible by the means of local networks, mesh networks or Bluetooth. We leave local and mesh networks for future work, and for our EpiMobile implementation, we have chosen Bluetooth, mainly because of its simplicity and availability in every device. More details on communication is in section 3.4.

4.4 Bioinformatic Analysis

We use Mash [30, 3] for metagenomic analysis (Section 2) in order to identify pathogens present in patient samples. While there exist other bioinformatic platforms for metagenomic analysis [15], we chose to use mash because of its small computational resource footprint (reported in depth in [30]).

At a high-level, mash enables rapid string matching between some query string (DNA sequence) and dictionary (reference database). Mash does this by decomposing a DNA string into a set of kmers (smaller and overlapping substrings of a pre-specified size) and applying the minhash [10] method to each kmer, which are then assembled into a single *sketch*. The distances between sketches are computed according to the Jaccard index [30], where 0 indicates a perfect match and 1 indicates complete dissimilarity. As an output mash will produce an ordered list of matching organisms (as represented by their genomic strings), from best match (Jaccard distance closer to 0) to poorest match, as well as the computed Jaccard distance, a p-value, and the total number of matching hashes (see `https://mash.readthedocs.io/en/latest/tutorials.html`).

The Jaccard distances can also be used to generate clusters of related organisms, however we do not use this functionality in EpiMobile. We use the pre-computed sketch of a reference database that mash provides, which contains approximately 54,000 organisms and is only 98 MB in size. New sketches can be easily added to the reference database if required without recomputing the entire database sketch.

Within EpiMobile, we run mash using default parameters with a kmer size of 16 (-k), a minimum of two copies per kmer (-m; to remove low quality sequences), and a sketch size of 400 (-s). We report the organism of the top mash hit, so long as the total number of matching hashes between the top hit and the query string was greater than 3, otherwise we report an `UNKNOWN` status. We arrived at this matching hash threshold by using an ebola MinION dataset(see Section 5) and observing the difference between the top hit and the subsequent hits in the file. Furthermore, we found that developing a threshold on the number of matching hashes, rather than Jaccard distance, improved EpiMobile's reporting accuracy across samples, especially as some samples were higher quality (generally Jaccard scores closer to 0) than others.

4.5 Workflow

Here we describe the workflow of EpiMobile to explain how we divided tasks among several modules and how they fit together in the entire workflow. We will also discuss the key communication concept that influenced some of our design decisions.

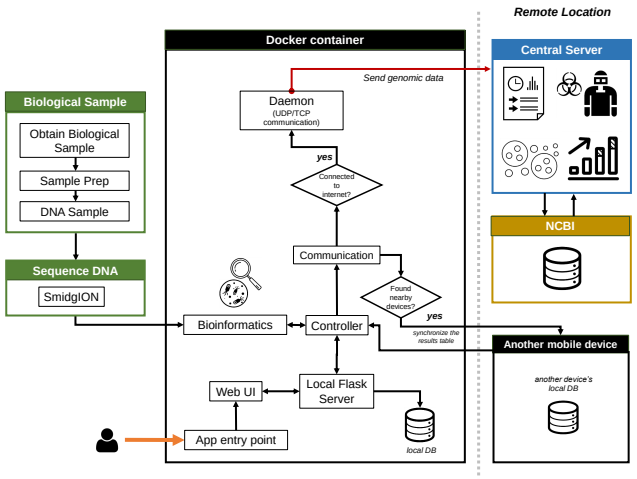

Figure 3 EpiMobile Implementation. The device input layer is represented by the biological sample and sequence DNA components. Data processing and application layers are contained within the large central Docker container component. Lastly the communication layer is represented by black (local) and red (global) lines leaving from the central docker container. EpiMobile's layers are described in Section 3.

Figure 3 shows the workflow of EpiMobile. It starts with a user launching EpiMobile application. Through a user interface, she will then see a dashboard displaying the results for previously analyzed DNA sequences and the new sequences that have not been processed yet, summarized in a high-level table of results (see figure 4). She will interact with the rest

of the system through a user-friendly web interface which is in turn connected to the Controller, the central module that connects all parts of the system together.

Upon a user request, the Controller calls the Bioinformatics module that performs genomics analysis on DNA sequences received from a MinION device. When analysis of a particular sequence finishes, the Bioinformatics module returns the result back to the Controller that stores it in a local Database.

Local files and results

Sequence ID	Highest Hits	Result
004674.2D.fastq.gz	Zaire_ebolavirus	Item already analyzed
SRR5121074_gi_1002634689_gb_KU853012.1_.fastq.gz	Zika_virus	Item already analyzed
014370.2D.fastq.gz	N/A	Analyze
015980.2D.fastq.gz	N/A	Analyze
013795.2D.fastq.gz	N/A	Analyze
SRR5197736_KX197192.1_PE243.fastq.gz	N/A	Analyze
015815.2D.fastq.gz	N/A	Analyze
013576.2D.fastq.gz	N/A	Analyze
015984.2D.fastq.gz	N/A	Analyze
013857.2D.fastq.gz	N/A	Analyze
013962.2D.fastq.gz	N/A	Analyze
014098.2D.fastq.gz	N/A	Analyze
014465.2D.fastq.gz	N/A	Analyze

Figure 4 Table of results as shown via EpiMobile web user interface

Communication module is responsible for data transfer between the device and a remote server and/or nearby devices. If the Communication module identifies that there is an internet connection, it will synchronize data with a remote server; in this case, we assume that the server was set up by the World Health Organization (WHO). The Communication module will synchronize the following:

- Local results table: to let the WHO know what the doctors working in the field are aware of;
- DNA sequences: for a more thorough and detailed analysis on specialized hardware and software;

Upon a user request, Communication module will turn on Bluetooth and synchronize local tables of results with its nearby devices. This is done for people working in the field to be able to see a more complete picture of a potential or ongoing disease outbreak. Communication component of EpiMobile is described in more detail in section 3.4.

5. EVALUATIONS

5.1 Resource Utilization Analysis

One of the main goals of this project is to propose a system that can be easily deployed on any reasonable machine; to confirm that, we have analyzed the EpiMobile resources utilization on a simple machine.

The Docker container itself is around 1.5 GB of virtual size, the blood analysis takes at most 600 MB of memory. For each patient sample analysis performed, $50K$ rows are inserted in the database; this is the output of the mash algorithm. It takes around 700 ms to analyze the patient sample against the reference database, totalizing 2.2 seconds to return the patient sample result back to the user, without the need of any Internet connection.

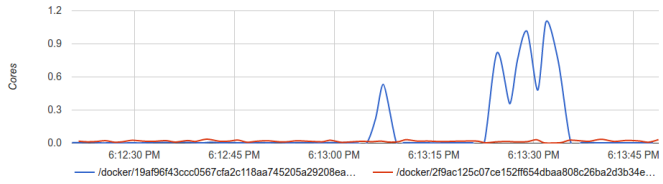

Figure 5 CPU usage when analyzing a patient sample

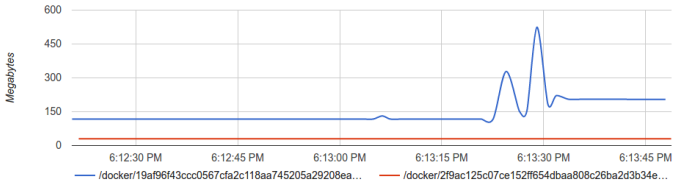

Figure 6 Memory usage when analyzing a patient sample

Repository	Tags	ID	Virtual Size	Creation Time
ubuntu	14.04	sha256:7c09e61e90350e8f5	179.26 MiB	2/27/2017, 11:41:06 AM
google/cadvisor	latest	sha256:f9ba08bafdeaf8158	54.69 MiB	3/9/2017, 3:30:29 PM
epimobile	latest	sha256:88f014991e675df10	1.54 GiB	4/17/2017, 5:53:59 PM

Figure 7: Docker container virtual size

5.2 Evaluation of Diagnostic Performance

We use standard diagnostic test descriptive statistics to evaluate the diagnostic aspects of our system. To derive these statistics we have created a synthetic dataset and assessed our bioinformatic components (Section 4.4) ability to *correctly* identify these organisms.

Evaluation Data. We used a dataset consistent of 20 ebola virus samples collected from patients [32] (`available at: https://github.com/nickloman/ebov`) and a collection of and 6 zika virus samples also collected from patients(obtained randomly from the following BioProjects: PRJNA358078 and PRJNA329512). Ebola virus samples were generated using the MinION platform and Zika Virus were generated using the Illumina (not portable) sequencing platform. Different genomic platforms will produce genomic data of different quality and so-called read length (Section 2).

Descriptive Statistics and Findings. We calculated the total number of instances when the EpiMobile system correctly classified (see Section 4.4) samples within our evaluation dataset. Currently, EpiMobile has 100% accuracy, but

this perhaps more of a reflection of MASH's performance rather than EpiMobile's. A small dataset might also underrepresent the performance and could result in skewed accuracy measures.

Applicability to real outbreaks. The data that we used in our analysis, especially the ebola and zika samples are representative of data collected, processed, and (genome) sequenced *in the wild* during active outbreaks. It is possible that these samples reflect lower quality sequencing data relative to what will be possible in the future. That is because in spite of talents and efforts of the investigators that derived these data, at the time these data were derived (2014 to 2016) MinION was still a relatively new technology. As such, these early studies were some of the first attempts to test MinION *in the wild* and more routine and standardized use in the future may yield higher quality data for analysis. Even though a synthetic data set is not an equivalent substitute for a prospective outbreak scenario assessment of EpiMobile's diagnostic capacity, our construction of the synthetic dataset thus offers a reasonable realistic suggestion of expected performance in future outbreaks.

One way in which our evaluation dataset does not reflect real sample data is that the authors of the ebola and zika datasets used sample preparation techniques to target and extract these organisms genomic data from patient's blood samples [32]. This is not quite a metagenomic approach. If we had used data that was processed through a truly bioinformatic approach, we would except that human DNA in the sample or pathogens from other chronic conditions (like HIV-AIDS or tuberculosis) would affect our ability to detect and correctly identify a newer acute pathogen infection. This means that our method is potentially overly optimistic relative to truly metagenomics approach.

6. ASSUMPTIONS AND LIMITATIONS

Our intention was to provide a conceptual sketch of the EpiMobile system and a minimal viable product implementation, and the validity and robustness of our results reflect this intention. However, as we are humble and poor grad students, there were also devices and software to which we did not have access. We detail those limitations in this section.

There are a few assumptions that we made when sketching the system and writing our implementation of the system:

- Currently we do not possess an access to a MinION device and therefore we cannot replicate the entire workflow of the EpiMobile system. We attempted to obtain, whenever possible, data files produced by the MinIon device. Further, since SmidgION is a conceptual product, we also cannot speculate on the details of its data quality or even the timeframe of its public release.

- Currently, Docker is not natively supported on any of the mobile operating systems, however there has been some community discussion to support this functionality in the future. It may be possible to run Docker on a phone is someone wished to hack through the installation processed, however, there is no official support yet. Due to this limitation, we assess out software performance on laptops running Unix, OSX and Windows and extrapolate our results to mobile phones.

- We also assume that once Docker is available for mobile phones, no significant changes will have to be made to EpiMobile.

7. USAGE SCENARIOS

In this section we describe the primary usage scenarios that our application supports. However, EpiMobile may be used for applications that we do not described here and may be modified to support expanded functionality in the future. We discuss some of that potential future functionality in these usage scenarios and broader applications in Section 9.

7.1 Initial Point of Care Diagnosis

Initial point-of-care diagnosis concerns the interaction between a clinician and the patient they are currently assessing. The main task in this interaction is to identify whether some underlying pathogen is the cause of a patient's symptoms and, importantly, which pathogen it is. We discuss two scenarios, one where a pathogen can be identified by EpiMobile, and another where the device cannot identify a pathogen. In both scenarios, the clinician will gather a sample from the patient (blood, urine, salvia, etc.), prepare the sample for analysis, and use the sequencing device (SmidgION, see Section 2.2.1), and finally run the EpiMobile application (Figure 8).

7.1.1 When an infectious agent can be identified

In this scenario, the EpiMobile application will provide a single response for the pathogen detected in the patient's sample, for example `Zaire Ebola Virus` (see Figure 4. If there is an Internet connection available, the mobile device will automatically transmit the genomic data to a central server for further analysis (see Section 7.3). If there is no Internet connection, the clinician can later co-ordinate her findings with her team (see Section 7.2). However, and most importantly, with the results on hand the clinician is able to make a more precise treatment decision for their patient. Knowing now that a patient has a potentially deadly and transmissible disease, the clinician can also initiate broader public health action to identify individuals that may have been in contact with the infectious patient and screen them for the presence of pathogen using EpiMobile. As the clinician sees, screens, and diagnoses more patients, results stored on her mobile device can make her aware of the number of patients harboring a specific pathogen. For now, she can assess on her own whether the number of new cases (individuals with the pathogen) is alarming, but in the future it may be possible for EpiMobile to make this assessment and alert her.

7.1.2 When an infectious agent cannot be identified

In some instances, EpiMobile may return an `UNKNOWN` result. This can occur for a number of reasons. One reason is that the concentration of pathogen genomic content in the sample is too low to be detected by the SmigdION platform. This can be remedied to some extent by changing the sample preparation procedure prior to using SimdgION, but this may not be easy to do in a clinical setting and as such we do not consider that a viable course of action. Linked to DNA content concentration is quality; a just barely sufficient concentration for detection may yield very poor quality genomic data making it difficult to identify an associated pathogen.

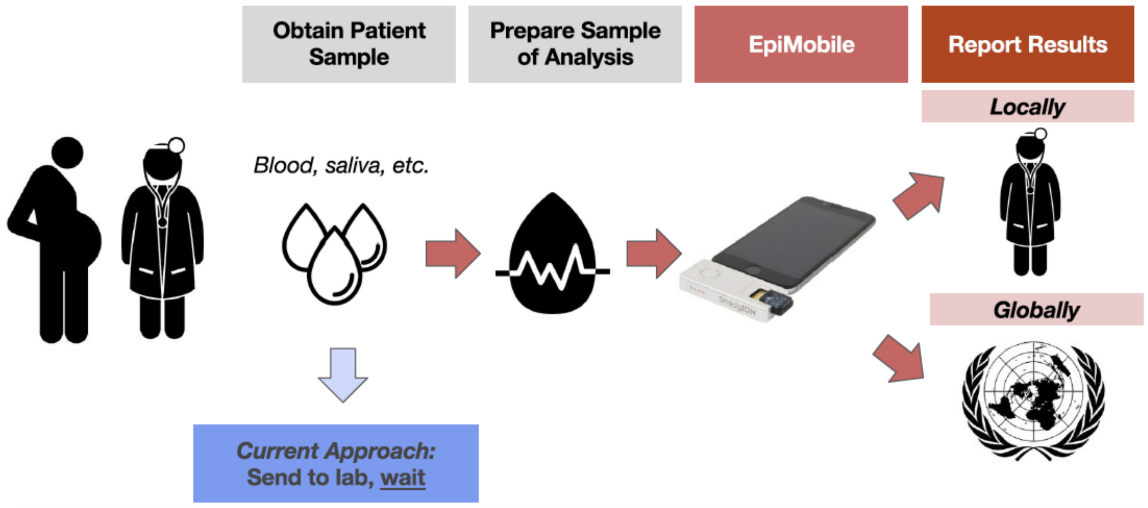

Figure 8 Summary of EpiMobile Usage. Currently, when a clinician obtains a patient sample she must send it to a central laboratory for analysis, which can take days or even weeks, depending upon the pathogen. With EpiMobile a clinician can get a result in real-time, sharing it with local clincial team members and, if an Internet connection is available, a national or global public health agency. *Figure icons are courtesy of the noun project. EpiMobile icon is SmidgION concept design [7]*

The theoretical limits on the devices change as the technology improves and with the type of sample collected. Assuming that there is enough of a DNA concentration for sequencing, and the quality is reasonable, another reason that a sample may not be detected is that the pathogen does not match one of the approximately 54,000 genomes in the mash database stored on the mobile device (Section 4.4).

EpiMobile's current approach is just to report the `UNKNOWN` status. If an Internet connection is available, the genomic content can be sent to the central server and more in depth analysis can be conducted. If a number of clinicians in a local area get an `UNKNOWN` status it is possible to launch an investigation into the cause.

Future work to handle EpiMobile `UNKNOWN` status could include more sophisticated on board quality assessment and, should DNA content pass quality control, the ability to integrate the genomic sequence processed by the device into the MASH database, which would allow for rapid and real time monitoring of a potentially emerging pathogen.

7.2 Local Co-ordination of Disease Status

EpiMobile's results are not confined to a single mobile device and can be shared in the absence of an Internet connection (Section 3.4.1). Clinical teams would specify when to co-ordinate and synchronize the information on their devices, namely tabular results that summarizes the pathogens detected on each device. With these summarized results on hand, the clinicians could discuss the case counts and plan more intensive public health actions if necessary. For example, if clinicians in one region could quickly identify a spike of Ebola cases, they may initiate more intensive quarantine protocols, alert local governments and agencies (with evidence and case counts on hand), and petition for more supplies. A clinician that leaves a field site to return to an area with an Internet connection may also share their stored tabular data with a central public health agency.

Currently with EpiMobile, local co-ordination of disease status only passes a simple table data structure between devices. It may be possible in the future to consider an alternative data structure that allows EpiMobile to facilitate more efficient sharing of genomic data as well. The local sharing of genomic data may be very important when new pathogens emerge for which there are no closely related entries in the mash database.

7.3 Global Co-ordination of Disease Status

When an Internet connection is available, EpiMobile will transmit the raw genome data analyzed by the mobile device to the a central server (Section 3.4.1). Here we presume that the central server belongs to an agency, for example the World Health Organization or a national public health agency, that is capable of analyzing the data more intensively, and that it is not just simply a storage server. With the sequenced results on hand, a public health agency can conduct more in depth phylogenetic and spatio-temporal analysis to monitor pathogen status nationally or even globally – analyses that are too intensive to (currently) realistically perform on a mobile device. Such an approach would constitute a passive surveillance system, where a limited amount of data is collected (usually just case counts) [21], that is used today by public health agencies to monitor disease – but EpiMobile would provide much faster, nearly real-time, and more in depth (due to genomic data) monitoring compared current approaches.

Currently, EpiMobile only incorporates `putting` the sequence information on a central server (although this feature is not yet full implemented), but in the future it may be possible to also `fetch` information from the central server, for example to validate results, to get updates for the mash databases or bioinformatic methods, and perhaps even contact specific clinicians directly.

8. RELATED WORK

Real-time or near real-time genomic sequencing. The MinION platform has been available to researchers as part

of an early access program launched in late 2013 [6] and the first published studies of its field application were during the 2014-2016 ebola outbreak [32] and the (currently) ongoing zika outbreak [17]. These studies used a laptop to initially process data from the MinION sequencer (a step that is not currently covered in EpiMobile because we cannot access that software) and would then rely on Internet connectivity to send and process samples on a cloud-computing service (see [32] supplemental materials). Comparing only to the evaluation component of [32] and [17] , EpiMobile attempts to perform analysis locally on the device as well as share results in the absence of an Internet connection, which enables a more rapid turn around time than in [32] and [17]. While EpiMobile supports a much more limited set of analyses, we argue that this is reasonable delineation between our more immediate clinical and public health applications and the research objectives of [32] and [17]. EpiMobile can be deployed to various laptop devices as well, thanks to our docker implementation. We found only one other suite of bioinformatics software designed for mobile devices, DNAApp [36], however it performs very different analytic functions that could not be used for diagnosis tasks and furthermore does not discuss data sharing. Outside of the MinIOn sequencing platform, there is also active research using the much larger, laboratory confined, sequencing technologies for rapid diagnosis, for example with tuberculosis [31] and Staphylococcus *aureus*, Clostridium *difficile* [16], and even rapid testing for antibiotic resistance genes in pathogens [9]. However, all of these studies, and many supporting bioinformatic pipelines not discussed here, rely on either cloud technologies or much more powerfully resourced compute clusters available to centralized agencies in predominately wealthy nations. This is to be expected as this area of research is evolving. However, this current state of the art also demonstrates the potential of EpiMobile to influence future directions of future genomics point-of-care work.

Other Biometric Sensors. Outside of genomics point-of-care, the routine collection, sharing, and analysis of biometric data through sensors, like FitBits or health apps on smartphones, for personal analytics is a growing application area. Currently, the most popular systems (pedometer and activity sensors) store and process much less data than EpiMobile and for potentially unknown health outcome benefits (reference: the authors personal experiences interacting with these devices) and as such we do not consider them much further.

The routine collection of biometric data is also being used in telemedicine or eHealth applications to provide healthcare services to rural, remote, or impoverished communities that have limited access to medical resources. Commonly, these systems will use sensors to collect types of data for a specific health outcome [39]. In one example authors created portable sensors that interfaced with smartphones for diagnosis sickle cell anemia [26], and while they perform some complex computations on the device they do not discuss the sharing of results. In another example sensors data was synced to a mobile device, a smartphone or laptop, before being synced to a central medical server [29] – a model similar to the one used by EpiMobile. Some simpler SMS based mobile eHealth applications have been used successfully in clinical trials to improve health outcomes of patients with HIV aids [12], and while this eHealth research shows the benefits of using mobile devices in health applications this application context differs from EpiMobile because it focuses more on two-way communication via SMS, and not more involved gathering and processing of data. We note that many of these examples focus on a clinical application and do necessarily consider the broader public health implications.

One challenge of these many different mobile device applications for health and medicine is that there is a lack of standards [39] and perhaps even a lacking common basis of research literature from which to draw upon. EpiMobile's implementation attempted to draw from bioinformatic and systems research knowledge as well as public health subject matter expertise to establish the system's conceptual design and implementation. Unlike some of the examples of systems discussed here, we focused much more heavily on communication of results beyond the analytic device.

Sensor Networks. Sensor network systems, which include smartphones, are designed to collect information from various sources, for example environmental measures like parts per million of air pollution, tracking wildlife, assessing traffic conditions, and lastly collecting biometric data from humans. Like EpiMobile the solutions developed for these sensor networks in various application contexts attempt to trade-off resource limitations and attempt to balance local processing on the device, for example the Hyrax [28] and mobileNet [22] or across multiple devices such as MagentOS [8], with global processing using cloud based services, or even some hybrid local/global model such as CloneCloud [11]. Compared to these systems, EpiMobile's implementation is far simpler, and constrained, in terms of data sharing and using other mobile devices for computation (although the system can evolve in the future, see Section 9.1). Finally, each of these different systems assumes different models (or topologies) of communication among sensors; whereas some systems limited communication between sensors as siphoning data through a central node in a star topology, other topologies support inter-sensor communication over peer-to-peer or two-tiered network topologies [27]. EpiMobile is designed to facilitate a peer-to-peer network topology for transmitting results, but it also sits in an interesting place among this literature due to the need for rapid turn-around time and sharing of results both locally and globally in the presence of network connectivity constraints. Thus, the application context of EpiMobile, and our solutions, presents some interesting constraints and trade-offs to the existing sensor network literature.

9. CONCLUSIONS

Genomics point-of-care is coming closer to being a reality with the advent of new portable genomic sequencing technologies. To anticipate the exciting opportunities in the future we have created EpiMobile, a conceptual prototype for genomics point-of-care workflows on a mobile device. EpiMobile's architectural designed attempts to minimize its computational resource footprint and facilitate local and global communication of results in the presence of Internet connectivity constraints. The design and implementation of EpiMobile can help healthcare teams mount more rapid responses to emerging or evolving disease outbreaks, whilst also communicating with national or global public health agencies whenever possible. These features of EpiMobile set it apart from current work that rely on cloud-

based data processing, or that process much less data on the device. While our findings are limited by the exploratory nature of our work and our lack of access to MinION and its proprietary software, we none-the-less believe that EpiMobile's design and specific diagnostic application context contribute an interesting and important basis for the future development of mobile device based genomics point-of-care applications.

9.1 Future Work

Currently, EpiMobile is a minimal viable product for pathogen point of care diagnosis. However, with this base in place there are a number of potential paths to explore as part of future work.

On the public health front is would be helpful to be able to also provide tailored treatment suggestions, and/or identify drug resistance when it occurs. However, treatment choices and responsiveness remains an open problem for the bioinformatics and molecular biology communities as clear links between genomic data, phenotypes (i.e. appropriate treatment), and outcomes (i.e. responsiveness to treatment) are difficult to establish [35]. Still for some pathogens where the link between genomic data and treatment is better known it may be possible to incorporate treatment suggestions.

There are also technical advancements that are possible for future version of EpiMobile. Alternative data models for efficiently transmitting and storing information, especially genomic data, could provide richer and more valuable information for clinical and local health teams to respond even more effectively. Fully distributed computing over mobile devices, for example using mobile ad networks, including mesh networks, could improve the sophistication of local analyses making EpiMobile even more robust and useful in resource constrained environments. More complex computations may also be possible as advances in artificial intelligence and machine are also being translated to mobile devices without relying on an Internet connection [25, 22].

10. ACKNOWLEDGMENTS

We would like to thank Prof. Andrew Warfield, who offered CPSC 538w course in the semester of Spring 2017, for teaching us how big distributed systems work, how to evaluate system performance, think critically and, last but not least, despise Java and its serialization absurdities :D

11. REFERENCES

[1] Distinctions between public health and medicine. https://www.hsph.harvard.edu/about/public-health-medicine/. Accessed:2017-04-20.

[2] Docker - build, ship, and run any app, anywhere. https://www.docker.com/. Accessed: 2017-03-29.

[3] Fast genome and metagenome distance estimation using minhash. https://github.com/marbl/Mash. Accessed: 2017-04-10.

[4] Flask (a python microframework). http://flask.pocoo.org/. Accessed: 2017-04-11.

[5] MinION: portable, real-time biological analyses. https://nanoporetech.com/products/minion. Accessed: 2017-03-26.

[6] Oxford nanopore expands minion early-access program as users present, publish early results. http://bit.ly/1D8DjKi. Accessed: 2017-04-20.

[7] SmidgION: nanopore sensing for use with mobile devices. https://nanoporetech.com/products/smidgion. Accessed: 2017-03-26.

[8] R. Barr, J. C. Bicket, D. S. Dantas, B. Du, T. W. D. Kim, B. Zhou, and E. G. Sirer. On the need for system-level support for ad hoc and sensor networks. *ACM SIGOPS Operating Systems Review*, 36(2):1–5, apr 2002.

[9] P. Bradley, N. C. Gordon, T. M. Walker, L. Dunn, S. Heys, B. Huang, S. Earle, . Pankhurst, D. Buck, G. McVean, A. S. Walker, T. Peto, D. Crook, and Z. Iqbal. Rapid antibiotic resistance predictions from genome sequence data for S. aureus and M. tuberculosis. *bioRxiv*, 6:018564, 2015.

[10] A. Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pages 21–29. IEEE Comput. Soc.

[11] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: Elastic Execution Between Mobile Device and Cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314, 2011.

[12] H. Cole-Lewis and T. Kershaw. Text messaging as a tool for behavior change in disease prevention and management. *Epidemiol Rev*, 32(1):56–69, 2010.

[13] F. S. Collins and H. Varmus. A new initiative on precision medicine. *The New England journal of medicine*, 372(9):793–5, 2015.

[14] X. Didelot, R. Bowden, D. J. Wilson, T. E. a. Peto, and D. W. Crook. Transforming clinical microbiology with bacterial genome sequencing. *Nature Reviews Genetics*, 13(9):601–612, 2012.

[15] M. Escalona, S. Rocha, and D. Posada. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nature Reviews Genetics*, 17(8):459–469, jun 2016.

[16] D. W. Eyre, T. Golubchik, N. C. Gordon, R. Bowden, P. Piazza, E. M. Batty, C. L. C. Ip, D. J. Wilson, X. Didelot, L. O'Connor, R. Lay, D. Buck, A. M. Kearns, A. Shaw, J. Paul, M. H. Wilcox, P. J. Donnelly, T. E. A. Peto, A. S. Walker, and D. W. Crook. A pilot study of rapid benchtop sequencing of staphylococcus aureus and clostridium difficile for outbreak detection and surveillance. *BMJ Open*, 2(3), 2012.

[17] N. R. Faria, E. C. Sabino, M. R. T. Nunes, L. C. J. Alcantara, N. J. Loman, and O. G. Pybus. Mobile real-time surveillance of Zika virus in Brazil. *Genome Medicine*, 8(1):97, 2016.

[18] G. M. Frampton, A. Fichtenholtz, G. a. Otto, K. Wang, S. R. Downing, J. He, M. Schnall-Levin, J. White, A. Sanford, [...], V. a. Miller, J. S. Ross, J. Curran, M. T. Cronin, P. J. Stephens, D. Lipson, and R. Yelensky. Development and validation of a clinical cancer genomic profiling test based on massively parallel DNA sequencing. *Nat Biotechnol*, 31(11):1023–1031, 2013.

[19] J. Gardy, N. J. Loman, and A. Rambaut. Real-time digital pathogen surveillance âĂŤ the time is now. *Genome Biology*, 16(1):155, 2015.

[20] S. Goodwin, J. D. McPherson, and W. R. McCombie.

Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet*, 17(6):333–351, 2016.

[21] L. Gordis. *Epidemiology, 5th edition*, volume 62. 2013.

[22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. apr 2017.

[23] M. Jain, I. T. Fiddes, K. H. Miga, H. E. Olsen, B. Paten, and M. Akeson. Improved data analysis for the MinION nanopore sequencer. *Nature Methods*, 12(4):351–356, 2015.

[24] M. Jain, H. E. Olsen, B. Paten, and M. Akeson. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biology*, 17(1):239, Dec 2016.

[25] J. Jin, A. Dundar, and E. Culurciello. Flattened Convolutional Neural Network for Feedforward Acceleration. *ICLR workshop*, (2014):1–11, 2015.

[26] S. M. Knowlton, I. Sencan, Y. Aytar, J. Khoory, M. M. Heeney, I. C. Ghiran, and S. Tasoglu. Sickle cell detection using a smartphone. *Scientific Reports*, 5:15022, 2015.

[27] J. P. Lynch. A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring. *The Shock and Vibration Digest*, 38(2):91–128, mar 2006.

[28] E. E. Marinelli. Hyrax : Cloud Computing on Mobile Devices using MapReduce. *Science*, 0389(September):1–123, 2009.

[29] A. Milenković, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications*, 29(13-14):2521–2533, 2006.

[30] B. D. Ondov, T. J. Treangen, P. Melsted, A. B. Mallonee, N. H. Bergman, S. Koren, and A. M. Phillippy. Mash: fast genome and metagenome distance estimation using MinHash. *Genome biology*, 17(1):132, 2016.

[31] L. J. Pankhurst, C. del Ojo Elias, A. A. Votintseva, T. M. Walker, K. Cole, J. Davies, J. M. Fermont, D. M. Gascoyne-Binzi, T. A. Kohl, C. Kong, N. Lemaitre, S. Niemann, J. Paul, T. R. Rogers, E. Roycroft, E. G. Smith, P. Supply, P. Tang, M. H. Wilcox, S. Wordsworth, D. Wyllie, L. Xu, and D. W. Crook. Rapid, comprehensive, and affordable mycobacterial diagnosis with whole-genome sequencing: A prospective study. *The Lancet Respiratory Medicine*, 4(1):49–58, 2016.

[32] J. Quick, N. J. Loman, S. Duraffour, J. T. Simpson, E. Severi, L. Cowley, J. A. Bore, R. Koundouno, G. Dudas, A. Mikhail, R. Ouédraogo, [...], K. Stoecker, E. Fleischmann, M. Gabriel, S. A. Weller, L. Koivogui, B. Diallo, S. Keïta, A. Rambaut, P. Formenty, S. Günther, and M. W. Carroll. Real-time, portable genome sequencing for Ebola surveillance. *Nature*, 530(7589):228–32, 2016.

[33] A. L. Rasmussen and M. G. Katze. Genomic Signatures of Emerging Viruses: A New Era of Systems Epidemiology, 2016.

[34] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. *Annual ACM Symposium on Principles of Distributed Computing*, pages 1–7, 1996.

[35] A. C. Schürch and W. van Schaik. Challenges and opportunities for whole-genome sequencing-based surveillance of antibiotic resistance. *Annals of the New York Academy of Sciences*, 1388(1):108–120, jan 2017.

[36] J.-Z. Sim, P.-V. Nguyen, Y. Zang, and S. K.-E. Gan. DNA2App: Mobile sequence analyser. *Scientific Phone Apps and Mobile Devices*, 2(1):2, dec 2016.

[37] T. Thomas, J. Gilbert, and F. Meyer. Metagenomics - a guide from sampling to data analysis. *Microbial Informatics and Experimentation*, 2(1):3, 2012.

[38] J. C. Wooley and Y. Ye. Metagenomics: Facts and Artifacts, and Computational Challenges. *Journal of Computer Science and Technology*, 25(1):71–81, jan 2010.

[39] E. D. Zubiete, L. F. Luque, A. V. M. Rodriguez, and I. G. Gonzalez. Review of wireless sensors networks in health applications. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1789–1793. IEEE, aug 2011.