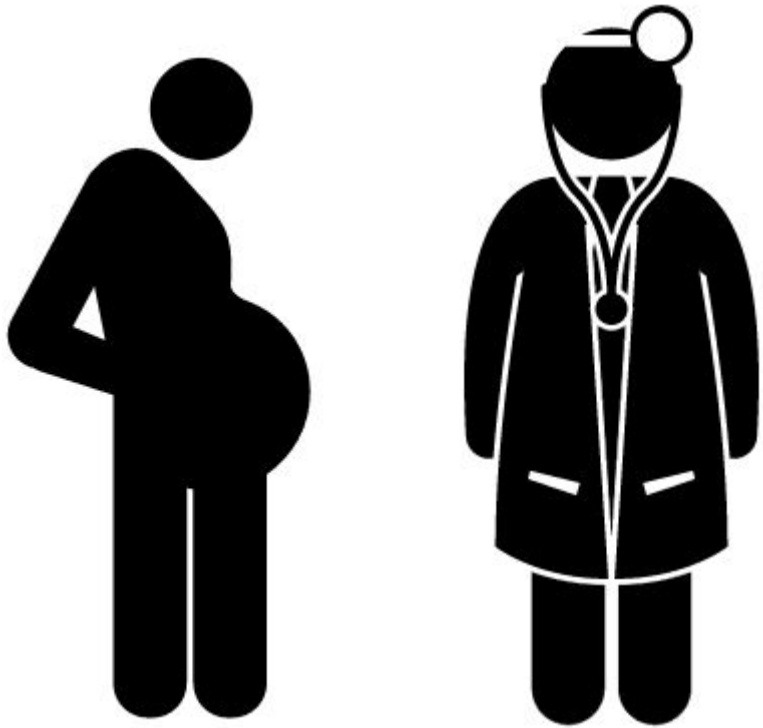
EpiMobile

Pathogen Point of Care Diagnosis and Global Surveillance using Mobile Devices

By: Anamaria Crisan, Rodrigo Araújo, Alexandra Kim

nagine the following scenario



Rethinking Disease Outbreaks

- **Time delays hinder rapid response to disease outbreaks**
  - *Limiting factor:* laborious and slow laboratory processes
- **Point of Care Genomics (POCG):** Detecting pathogen (harmful infectious agent) genomic content from human samples (blood, saliva, other fluids)
  - Technological advancements have made this possible
- **Near future opens POCG possibilities on mobile devices**
  - More versatile, can be used in clinics & in the field
  - Obtain real-time results, change course of public health response
- **Many technical & computational challenges remain for POCG on mobile devices**

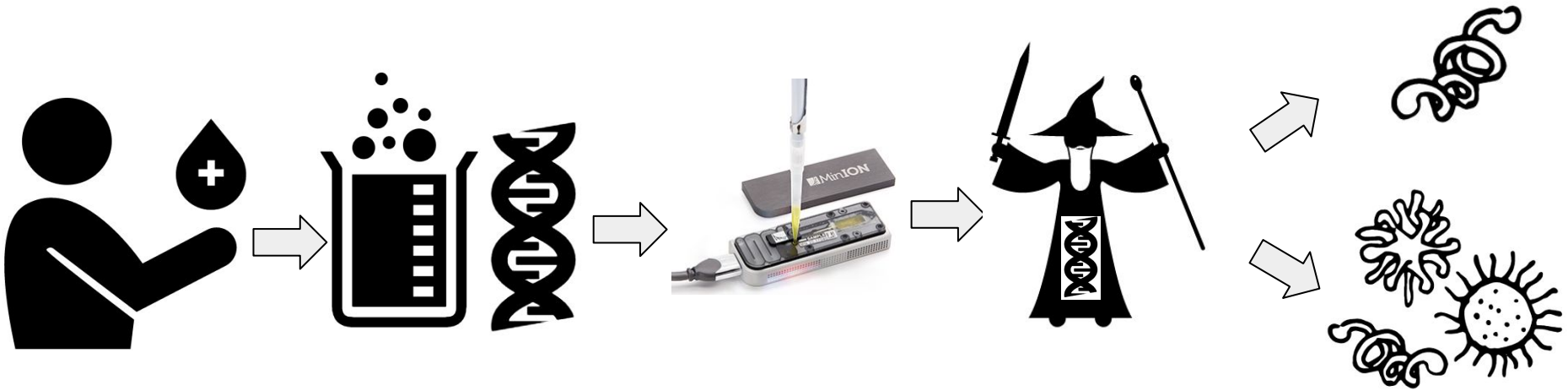
Genome Sequencing : A Changing Landscape

Genome Sequencing : A Changing Landscape

The Future: SmidgION

Genome Sequencing : A Very Brief Primer

Obtain Patient Sample	Isolate DNA	Sequence DNA	Bioinformatic Analysis	Resolve Genome



Who's there?

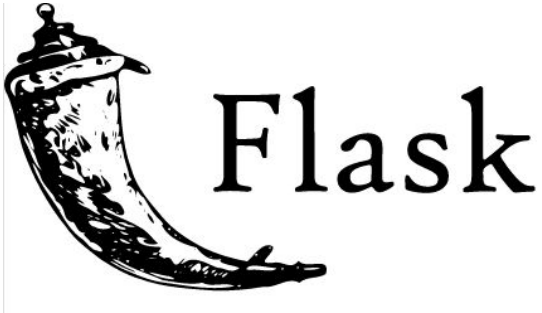
Challenges of Mobile Genomics Point of Care

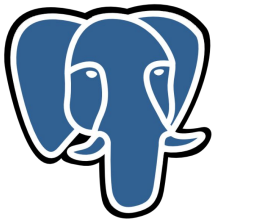
- **Computational resources constraints**
  - Processing genomic data requires a lot of computational resources
  - Compute + memory of phone increasing, but still there are limits
- **Bioinformation software dependencies**
  - Open-source software with various dependencies
  - Either redesign software for phones, or find another work around
- **Communication resource constraints**
  - In the field, internet access may not be consistently available
  - May delay data analysis if reliant on internet connection
- **Key question (for us):** What analysis can be done locally (on mobile device) and what must be done globally (using remote server)

Contributions

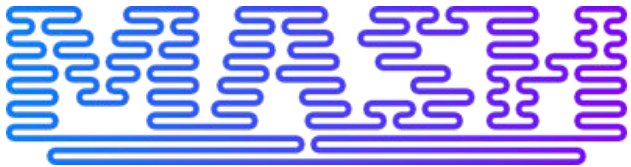
- Describing an end-to-end system that identifies pathogens from blood and returns the result on a mobile device and a central public health agency.
- Exploring cross-layer interactions between the processing, analysis and reporting layers on a mobile device, involving communication to cloud services.

Implementation: Stack





PostgreSQL



Implementation: Bioinformatics algorithm

What we use: Custom Python Script, MASH software

Main idea: String matching operation: find, and quantify distance, of matching genome within a reference database (dictionary) using MASH algorithm

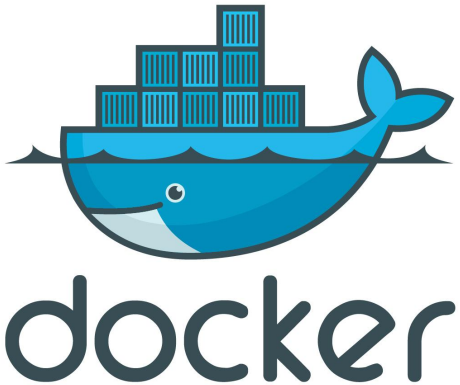
Report top match only if top match passes minimal distance threshold, else return “Unknown”

Sample genomic material & reference DB formatted for MASH algorithm

Implementation: Software container with Docker

Main idea: To be able to run the system across many devices

Docker is an open-source software container engine that enables separation of the application code from the underlying infrastructure. It runs an application in an isolated environment thus alleviating the need to write different versions of the system.



Implementation: Local vs. Global Communication

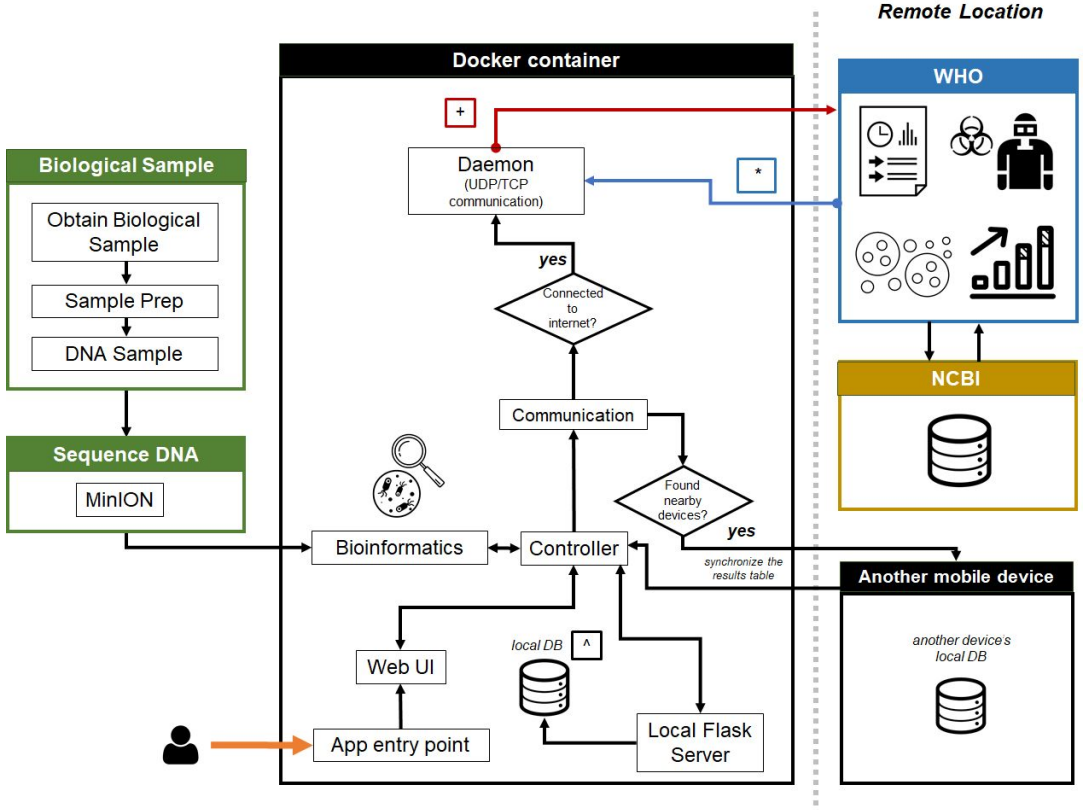
Main idea: Minimize transfer of data.

Local communication: after we process a sample using mobile device, it summarizes results into a table, which is communicated locally to other devices.

Global communication: if internet connection is available, send genomic data to the cloud (remote server) for verification and more in-depth analysis.

System Architecture

1) Obtain a biological sample
2) Get a processed DNA sequence from a MinION
3) Process the sequence on a device and get the results
4) Store the results in a local DB
5) Upon having Internet connection
   - each phone will send obtained DNA sequences to a remote server
   - phones will synchronize data with the remote server and send the local results table to the server
6) Upon a user request, synchronize the results *(not sequences!)* with other mobile devices via a local net

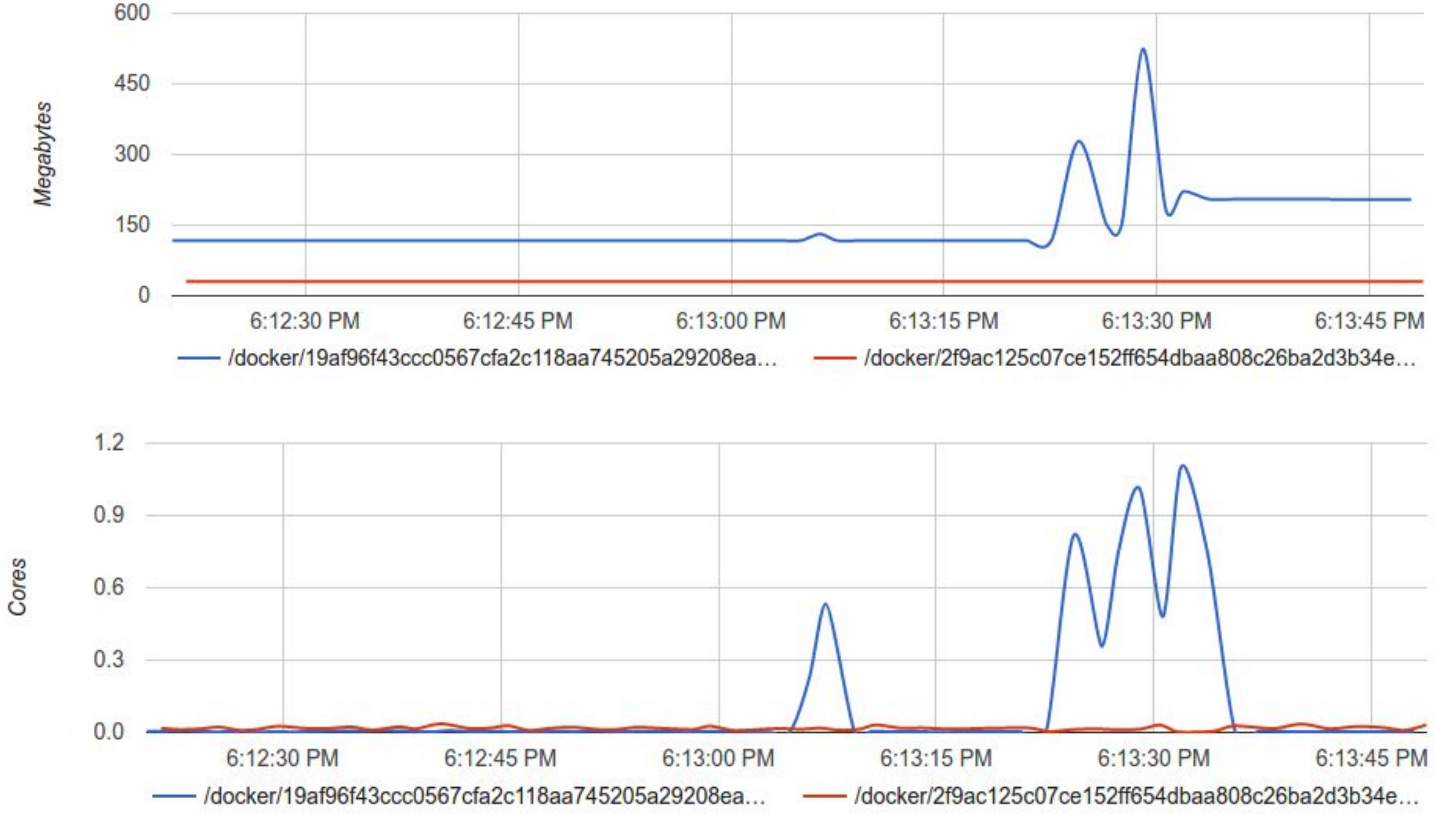


Revisiting Constraints & How They're Addressed:

- **Computational resources constraints**
  - Rapid processing using "good enough" first pass (MASH)
- **Bioinformation software dependencies**
  - Docker containers
- **Communication resource constraints**
  - Share data tables over bluetooth when internet is not available
  - Combined computational & human data sharing protocols
  - Pass genomic data to remote central server for more thorough analysis / verification when internet is available

Demo ;)

Evaluation - Performance analysis



Evaluation - Performance analysis

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

Some interesting numbers:

- **50k** rows written to postgresql instance for *each blood sample*
- **0.7 seconds** to analyze a blood sample *(4mb)* against the ref db *(94mb)*
- **2.2 seconds** to return sample result to the end user

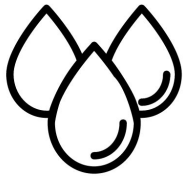
Evaluation - Diagnosis Descriptive Statistics

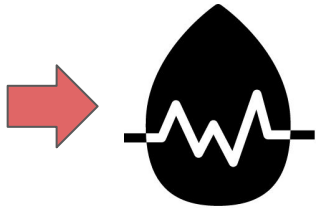
- **Evaluation dataset:**
  - 20 Ebola Virus Samples; 6 Zika virus samples
  - Ebola Virus sequence with MinION from recent guinea outbreak
  - Zika from various countries, sequenced with different platform (not MinION)
- **Classification criteria:**
  - Top match + minimum distance threshold
  - If doesn’t meet distance threshold : Unknown
  - Optimized for Ebola Virus data
- **Results:**
  - So far, 100%
- **Limitations:**
  - Sample preparation technique targeted Ebola virus, may over-represent accuracy

Usage Scenario Revisited

Obtain Patient Sample | **Prepare Sample of Analysis** | **EpiMobile** | **Report Results**

Blood, saliva, etc.





Locally

Globally

Current Approach:
Send to lab, wait

Assumptions and Limitations

- We run our analysis on the preprocessed data that can be obtained from MinION (that we don't have access to MinION ~~*because we are poor grad students*~~)
- To run the application on a mobile phone, SmidgION will be needed
  - SmidgION a prototype
  - It is still possible to run the app on a laptop, using MinION
- Docker is not natively supported on Android but will be in the future (hopefully)
- Docker container tested on Unix, OSX and Windows machines
- Once Docker is supported on Android and/or iOS, no complex changes will have to be made in the codebase

Conclusions

- EpiMobile: a system architecture scheme for point of care diagnosis and global surveillance on mobile devices.
- Future work:
  - Implementing global communication part
  - Porting the system to mobile phones
  - Implementation of the system involving an actual MinION device
  - Explore the possibility of using other local networks instead of Bluetooth

Thank you!