doi:10.20944/preprints202001.0107.v1

Ten recommendations for supporting open pathogen genomic analysis in public health settings

Allison Black[1,2], Duncan R. MacCannell[3], Thomas R. Sibley[2], Trevor Bedford[1,2], *on behalf of* The Public Health Alliance for Genomic Epidemiology (PHA4GE)

1 Department of Epidemiology, University of Washington, Seattle, Washington, USA.

2 Vaccine and Infectious Disease Division, Fred Hutchinson Cancer Research Center, Seattle, Washington, USA.

3 Office of Advanced Molecular Detection, Centers for Disease Control and Prevention, Atlanta, Georgia, USA.

Abstract

Public health agencies are increasingly using pathogen whole genome sequencing (WGS) to support surveillance and epidemiologic investigations. As access to WGS has grown, greater amounts of molecular data have helped improve our ability to detect outbreaks, investigate transmission chains, and explore large-scale population dynamics, such as the spread of antibiotic resistance. However, the wide adoption of WGS also poses challenges due to the amount of data generated and the need to transform raw data prior to analysis. This complexity means that public health agencies may need more advanced computational infrastructure, a broader technical workforce, and new approaches to data management and stewardship. As both a guide for how this development could occur, and a place to initiate discussion, we describe ten proposals for developing and supporting an informatics infrastructure for public health.

doi:10.20944/preprints202001.0107.v1

Recommendations

1. Support data hygiene and interoperability via development and adoption of a consistent data model

Public health programs currently lack a widely adopted, pan-pathogen, genomic data structure. This limitation leads to variability in data reporting and likely contributes to data incompleteness. Programs also lack consistent data models and ontologies for related epidemiologic data. Data models should provide sufficient structure to standardize reporting, while remaining flexible enough to be applicable across a wide array of pathogens.

2. Strengthen application programming interfaces (APIs)

Good APIs are critical to making data accessible and shareable. Existing frameworks, such as the General Services Administration's (GSA) API standards, can be an invaluable tool in the development of interoperable systems and services for public health. At a minimum, APIs should be RESTful [1], versioned, consistent, and have detailed documentation with working examples describing expected behavior. Where possible, APIs for commonly used workflows, components, and services should be standardized based on community consensus to maximize interoperability and flexibility of bioinformatic software.

3. Develop guidelines for management and stewardship of genomic data

These guidelines should describe which data to archive (including both raw and assembled data types), the duration of archival, systems for long-term archiving, the intended use and long-term value of the data, and appropriate metadata standards. Archiving practices must be responsive to local requirements and priorities, but where possible, they should always prioritize keeping data easily searchable and shareable.

4. Make bioinformatic pipelines fully open-source, broadly-accessible, and transparent

While commercial off-the-shelf software may provide turnkey bioinformatic capabilities to laboratories with limited bioinformatics capacity and IT resources, emphasis on the development and adaptation of open-source software for public health bioinformatics and data management is crucial for the advancement of global public health. Where possible, bioinformatic pipelines should be built around open-source software and be deployed openly so that expensive proprietary software licenses do not limit access to genomic assembly, analysis, and comparison. Graphical user interfaces should be developed for frequently used pipelines to lower the technical barrier to bioinformatic processing. These pipelines should output to common file formats, such as JSON, CSV, and FASTA. Using standardized data formats and structures will ease interaction with APIs, thereby supporting data sharing, analysis, and archiving efforts.

5. Develop and support pipelines for data visualization, exploration, and automated analysis

While the inferences gained from genomic data can be very important, the interpretation of genomic data is not always intuitive. This can create difficulties in communicating genomic findings to multidisciplinary public health teams as well as to the general public. Thus, data visualization and exploration tools are increasingly vital to support data interpretation and to communicate key findings. Analytic and visualization pipelines can be automated, improving the speed with which analyses can be run and interpreted, and supporting the reproducibility of results. Where possible, these visualization tools and pipelines should be separated from bioinformatic analysis pipelines to improve modularity and prevent unnecessary rerunning of computationally intensive tasks.

6. Improve the reproducibility of bioinformatic analyses

Laboratory assays that support infectious disease diagnostics, surveillance, and clinical decision-making are subject to rigorous validation and quality assurance protocols. These frameworks for accreditation and quality management will eventually extend to include many of the routine bioinformatic analyses used in public health. The need for reproducibility should drive how bioinformatic pipelines are developed, maintained, hosted, and tested. Developers should prioritize software stability and version control, containerizing code and requirements. Full pipelines and datasets should also be version controlled. End-to-end proficiency testing could be extended beyond wet lab protocols to include assessment of bioinformatic assembly against known standard datasets. Finally, workflow management software can be used to help support reproducibility.

7. Use cloud computing to improve the scalability and accessibility of bioinformatic analyses

As the scope of genomic surveillance grows, so too will the volume and complexity of data generated during routine public health laboratory operations. For many public health institutions, WGS data management already falls well within the realm of big data, and the assembly and analysis of next-generation sequence data is increasingly dependent on advanced computing infrastructure for data capture, analysis, and storage. Investments in capital infrastructure, as well as a specialized IT and bioinformatics workforce, are becoming increasingly important for strategic and budgetary planning at many institutions. Cloud computing greatly improves access to robust and scalable high-performance computing infrastructure, allowing users to scale up elastically during times of high demand, such as outbreaks, and scale down when demand is low to save costs. Workflow management software will be critical for supporting scalability across infrastructures and for supporting hybrid computing environments.

8. Support new infrastructure and software development demands with technical personnel

As public health agencies generate and analyze more complex data, develop more custom software and workflows, and move computing from the bench to the datacenter and into the cloud, there is growing recognition that new technical specialties will be needed to staff tomorrow's public health workforce. The transition of PulseNet and other large-scale public health surveillance programs to WGS has underscored the importance of workforce competency in bioinformatics, scientific computing, and data science. Given their expertise in analyzing and visualizing high dimensional and genomic data, bioinformaticians and data scientists will be critical to support a growing number of public health surveillance activities with exploding data inflow and sophistication. Furthermore, particularly for larger institutions that host software and contribute to development, the workforce should be expanded to include teams with software engineering experience.

9. Improve the integration of genomic epidemiology with traditional epidemiology

Neither epidemiologic case data nor pathogen genomic data are as powerful on their own as they are when they are integrated and analyzed in a timely and actionable manner. This integration will require technical expertise to develop secure systems for joining epidemiologic and genomic databases; new tools, algorithms, and ontologies to address missing and poorly-structured or complicated data sources; and strategies to ensure appropriate encryption and access control to combined datasets and personally-identifiable information. Technical challenges aside, effective integration of epidemiologic and laboratory data will also require frequent and open communication between surveillance epidemiologists, bioinformaticians, and other individuals conducting sequence analysis. To facilitate communication, agencies should consider providing traditional epidemiology training to bioinformaticians and genomic data analysis and interpretation training to epidemiologists.

10. Develop best practices to support open data sharing

Public health agencies may benefit from guidelines that describe which genomic data should be shared, which metadata should be shared, which platforms should be used for data sharing, and over what timescales data release should occur. To retain the usefulness of genomic data it is important that the analytic value of metadata is appropriately recognized and balanced alongside identifiability concerns.

Introduction

Increasingly, public health officials are using pathogen genomic sequence data to support surveillance, outbreak response, pathogen detection, and diagnostics. Sequencing cuts across traditional pathogen boundaries; for example, we can use it to distinguish cases of wild polio from vaccine-derived polio, or to predict the susceptibility of a tuberculosis infection to antibiotics, or to trace the source of a foodborne infection. Because of its utility, public health agencies throughout the world are developing their capacity to perform genomic sequencing. However, with new data streams come new challenges; next generation sequencing (NGS) data must be transformed from its raw state to be interpretable, and many of the tools developed for sequence assembly and analysis are either expensive to license or require a high level of computational proficiency to use. This means that the capacity to perform genomic data assembly and analysis has not expanded as widely as the adoption of sequencing itself. In this paper, we describe the current challenges that public health agencies face in supporting bioinformatics and genomic epidemiology, and provide proposals for building a sustainable infrastructure that can be used across public health programs.

To investigate the current landscape of bioinformatics and genomic epidemiology in public health agencies, we conducted a series of long-form, semi-structured interviews with bioinformaticians, laboratory microbiologists performing sequencing, software engineers developing pipelines and workflow management software for public health, and epidemiologists acting upon inferences from genomic data. We aimed to get a broad perspective, interviewing individuals from different countries, working on a wide array of pathogens, and working in agencies with varied capacities for genomic analysis. The interviews focused on the following topics: (1) technical components of genomic analysis, including data analysis goals, available compute infrastructure, frequently used software, and data archival practices; (2) considerations for genomic analysis specific to public health settings; and (3) social issues surrounding genomic data, such as concerns around data sharing and governance, and the integration of genomic and traditional epidemiology.

We hope that these proposals can help guide the development of a software ecosystem that is:

Reproducible, such that genomic analysis is standardized and repeatable;

Accessible, both at varying levels of economic resources and of technical knowledge;

Flexible, providing a set of modular tools to analyze, explore, and visualize genomic data across a range of public health applications; and

Auditable, ensuring that pipelines and their reproducibility can be validated according to strict public health standards.

The proposals outlined here inherently build off of one another; they are not a checklist. Rather, we aim to provide a structured view of what a public health informatics ecosystem might look like. We hope this work provides a starting point for others to join in helping to shape and build this ecosystem.

doi:10.20944/preprints202001.0107.v1

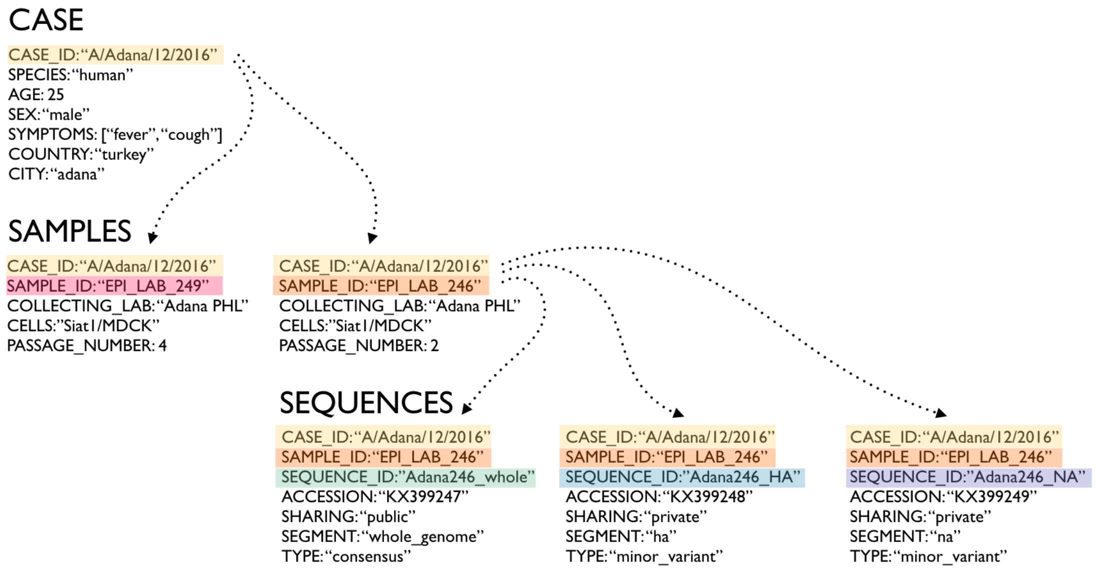
1. Support data hygiene and interoperability via development and adoption of a consistent data model

The value of an isolate is dictated not just by its molecular characteristics. A sample needs context; who was the sample collected from, when was it collected, how was it collected? Without this information much of the value of the sample is lost, both from clinical reporting and data analysis standpoints. Despite the value of metadata, sequence records are frequently decoupled from the full constellation of epidemiologic data describing the sample. While there are various reasons why this occurs, breakdowns in data hygiene compromise the utility of the data and impact users' ability to interact with the data programmatically. This limitation poses a critical issue, as increasing amounts of data reduce public health officials' ability to manually interact with the data. Complete and structured data is fundamental to an informatic ecosystem that can work at scale.

This is a commonly recognized problem, and widely used genomic databases such as the Sequence Read Archive (SRA) have standards and formatting requirements for submissions that do improve data hygiene. However, there is still much room for improvement, especially because many sequences can map to a single set of epidemiologic data in a way that is complex and often hierarchical. As an example, imagine all the possible sequences that could be collected from a single individual infected with influenza. Patient demographic data, clinical data, and exposure information form an unchanging set of characteristics describing the individual at that point in time. However, that sample may yield many distinct pieces of genomic data. For instance, scientists can ascertain the consensus genome of the infecting strain by sequencing the clinical isolate. From that same dataset they may also have separate SNP calls describing within-host minority variants. Additionally, a lab may decide to culture the infecting strain and sequence the cultured isolates after different numbers of passages. These scenarios yield sequences that are distinct from one another, and have different laboratory-associated data, but that share the same epidemiologic data. Ideally, the public health community would have a data structure and standardized vocabulary that accommodates and organizes these different data fields, linking them appropriately and annotating them consistently. We propose that the public health bioinformatics community, along with engagement from the major data repositories (e.g. NCBI, EBI, and DDBJ), develop common data models and adopt ontologies. To initiate this discussion, we describe an example of a data model below, and list ontologies that could be adopted more widely.

A data model for hierarchical genomic data.
As one example, the Nextstrain platform [2], employs a custom database that canonicalizes pathogen genome data and associated metadata. Data are sourced from a variety of public databases (NCBI, GenBank, and ViPR) as well as from GitHub repositories if permitted by the owners. Because the data are pulled from various sources, different types of data are provided and data are often varied in format. Thus, the agglomerated dataset must be standardized according to a schema that can apply to all sequences. The Nextstrain data model (Figure 1) includes three major data fields: case, sample, and sequence. Each case record can have multiple samples, and each sample can have multiple sequences. Linkage between the fields is maintained by a case identifier and sample identifier that are logged as subfields. Within each field, subfields are tailored to record the most pertinent information to that field. For example, the case field

contains information about host species, age, sex, symptoms, collection date, and geography. The sample field contains the case identifier, along with relevant information about the sample, such as collection date, collection medium (blood/urine/tissue), and culture information such as the cell line used and the number of passages that a sample underwent. This field also contains information about the lab that grew or tested the virus; this lab is often distinct from the clinic where the case presented. Finally, each sample may have multiple sequences. The sequence field specifies the case identifier and the sample identifier, but again organizes information more pertinent to the sequences themselves, such as (1) what portion of the genome the sequence is from, (2) whether the sequence is a consensus sequence or a minor variant, (3) flags specifying whether the sequence is public or private, and (4) accession numbers if the sequences were pulled from a public database.



CASE_ID:"A/Adana/12/2016"
SPECIES:"human"
AGE: 25
SEX:"male"
SYMPTOMS: ["fever","cough"]
COUNTRY:"turkey"
CITY:"adana"

CASE_ID:"A/Adana/12/2016"
SAMPLE_ID:"EPI_LAB_249"
COLLECTING_LAB:"Adana PHL"
CELLS:"Siat1/MDCK"
PASSAGE_NUMBER: 4

CASE_ID:"A/Adana/12/2016"
SAMPLE_ID:"EPI_LAB_246"
COLLECTING_LAB:"Adana PHL"
CELLS:"Siat1/MDCK"
PASSAGE_NUMBER: 2

CASE_ID:"A/Adana/12/2016"
SAMPLE_ID:"EPI_LAB_246"
SEQUENCE_ID:"Adana246_whole"
ACCESSION:"KX399247"
SHARING:"public"
SEGMENT:"whole_genome"
TYPE:"consensus"

CASE_ID:"A/Adana/12/2016"
SAMPLE_ID:"EPI_LAB_246"
SEQUENCE_ID:"Adana246_HA"
ACCESSION:"KX399248"
SHARING:"private"
SEGMENT:"ha"
TYPE:"minor_variant"

CASE_ID:"A/Adana/12/2016"
SAMPLE_ID:"EPI_LAB_246"
SEQUENCE_ID:"Adana246_NA"
ACCESSION:"KX399249"
SHARING:"private"
SEGMENT:"na"
TYPE:"minor_variant"

Figure 1. This schematic illustrates the data model used by the Nextstrain-Sacra database.

Ontologies for genomic epidemiology.
If widely adopted, the use of a common data model, or a set of common data models, will provide a unified framework for linking sequence data, clinical data, and epidemiologic information. However, to fully structure these data, public health programs will also need to adopt and/or develop ontologies that standardize free-form epidemiologic information about cases and their exposures. Two good examples of epidemiologic ontologies are IRIDA's genomic epidemiology ontology, GenEpiO, and FoodON [3]. These ontologies create controlled, standardized vocabularies, which facilitate programmatic interaction with databases and enable users to automate quality control and analytic procedures.

2. Strengthen application programming interfaces (APIs)

Application programming interfaces, or APIs, are the mechanism by which users communicate with computers, code, and databases in an automated way. They are critically important for programmatic querying of databases, collation of disparate data sources, and communication between pieces of software within a greater ecosystem.

The relative paucity of consistent and well-documented APIs for software tools and databases affects public health bioinformatics in at least two ways. Firstly, the lack of APIs limits the scalability of bioinformatic analyses. Currently, querying genomic databases frequently requires human interaction via a web-based graphical user interface. However, with ever increasing amounts of data, the ability to manually explore, source, and distribute data will decline. Agencies will need to have tools for automated querying and communication, and the quality of APIs will directly affect the ease with which public health programs can perform these functions reproducibly and efficiently. Secondly, the lack of APIs leads to inefficient use of bioinformatician effort. When basic pipelines do not run automatically, or significant effort is required to link programs together into a pipeline, bioinformaticians spend large amounts of time writing interstitial code and managing file format conversions. Some interviewees noted that performing these tasks reduced their availability to do more sophisticated genomic analyses that might have greater public health utility.

The development and use of well-documented APIs will underlie the success of a software ecosystem within public health and cannot be an afterthought. Public health institutions should adopt API standards, and APIs should be developed in tandem with database or software development. For the many software programs and databases that already exist, specific funding sources should be allocated to build or extend current APIs to function with the agreed-upon data models and adhere to adopted API standards.

Within the United States, the GSA develops and publishes APIs to provide open government data in machine-readable formats. As part of this effort, the GSA has developed standards for APIs. These standards provide a concrete starting point in the development of APIs for genomic and epidemiologic databases, and are described in detail at github.com/GSA/api-standards. In brief, the standards provide the following guidelines:

(1) APIs should be RESTful, with clear, human-readable endpoints.

(2) They should return JSON objects for both API responses and error messages.

(3) APIs should be versioned, and they should be backward-compatible within a major version, though breaking changes can occur with major version changes.

(4) The API must have clear and readable documentation and allow users to report feedback or issues and ask questions.

(5) All APIs should use HTTPS.

doi:10.20944/preprints202001.0107.v1

3. Develop guidelines for management and stewardship of genomic data

The increasing abundance of longitudinally collected pathogen genomic sequence data is a valuable resource for public health. However, to fully realize the value of this data, programs will need to manage and care for the data in a unified manner. To this end, public health institutions should develop and adopt guidelines and standards for data collection, annotation, archival, and reuse. These guidelines should be designed to ensure that data adhere to FAIR principles, such that archived data are **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable [4]. These principles and how they relate to genomic data are summarized below. Greater elaboration on data FAIRness is available in [4].

> FINDABLE: Data observations need globally unique identifiers that are persistent. Data observations need to have rich descriptions of the data, including how the data were obtained. These observations would be specified by the data model discussed in Proposal 1. Data observations should be indexed.
>
> ACCESSIBLE: Data should be retrievable using automated protocols that are open-access and universally implementable, and that provide authentication and authorization procedures. This principle is primarily met by the use of well-designed and documented APIs, as described in Proposal 2.
>
> INTEROPERABLE: Data are represented with standardized vocabularies (ontologies) that also follow FAIR principles. This ensures that data are accessible and shareable across agencies.
>
> REUSABLE: Data need to be richly described with relevant attributes about how they were generated and their provenance. Data released into open databases should have clear licenses describing how others can use the data.

Some of these principles are already being followed. The majority of agencies that we interviewed regularly submit raw sequencing reads to the SRA, which both archives the data for future use and publishes the data so that they can be found by other public health practitioners and scientists. Saving the raw data has the additional benefit of enabling users to completely re-run pipelines from start to finish, allowing comparison and beta testing of pipelines and analyses while protecting users from irreparable analytical mistakes [5].

While the consistent submission of raw reads to the SRA is an excellent first step, the community needs a formalized data stewardship framework that encompasses assembled genomic data as well. To promote findability and reusability, genomic assemblies should be annotated with information about how the assembly was made, such as what reference genome was used for mapping and what pipeline version was run. Submitted datasets should also be versioned. To ease the burden of archiving both raw reads and assemblies, submission mechanisms should be automated and fully integrated with bioinformatic assembly pipelines.

For further consideration of how to best preserve genomic data and curate high quality databases, please refer to Goodman et al (2014): Ten Simple Rules for the Care and Feeding of Scientific Data [6], and Hart et al (2016): Ten Simple Rules for Digital Data Storage [7].

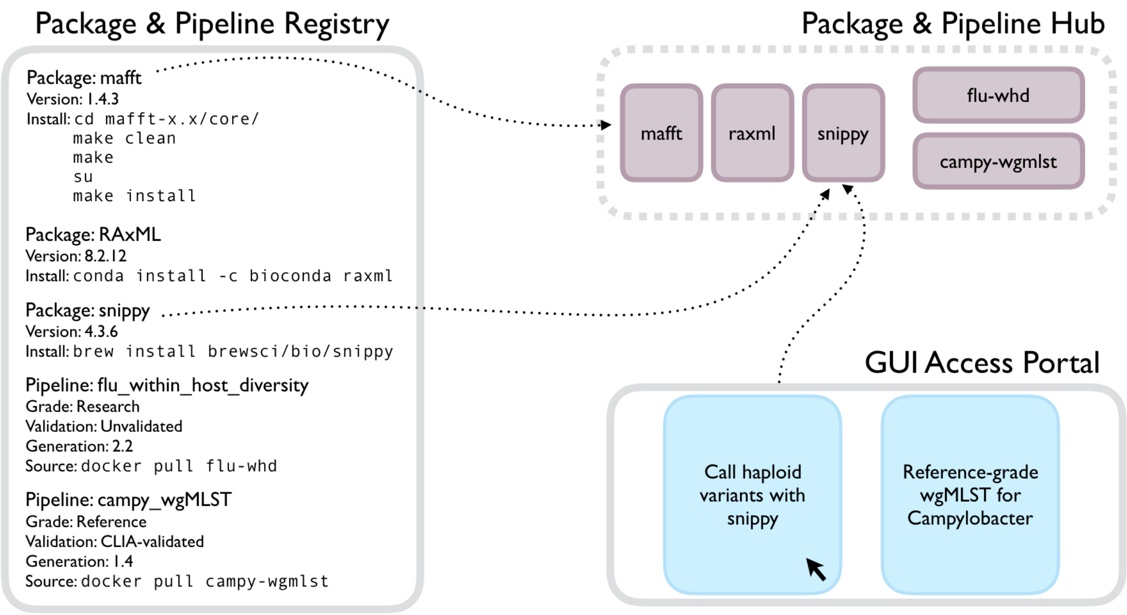
4. Make bioinformatic pipelines fully open-source, broadly accessible, and transparent

Currently, one of the most frequently used software platforms for bioinformatic analysis in public health laboratories is BioNumerics (Applied Maths/bioMérieux). This commercial software package has played a central role in the development of PulseNet's laboratory-based foodborne bacterial disease surveillance for more than 20 years. BioNumerics provides a bioinformatic analysis toolkit, but more importantly acts as a national database with specimen and process tracking and within-network data sharing. While many of the agencies that we interviewed appreciated the ease of use of the workflows and the integrated databasing capabilities of BioNumerics, multiple agencies also reported instances where the lack of modularity limited options for custom development and expansion, or where licensure costs impacted their ability to access the software. While BioNumerics provides developer resources, the field would benefit from moving towards a public health software ecosystem that supports the development, adoption, maintenance, and hosting of a set of reference pipelines that are openly developed and completely accessible.

Ensuring that pipelines are accessible will require developers to consider pipeline users, who within public health may have limited to no formal training in bioinformatics. Currently, many public health agencies are undergoing a transition period in which they can generate large amounts of sequence data, but have more limited access to the technical expertise needed to assemble and analyze the data. We found that this need was generally harder to meet in smaller and lower-resource settings. Interviewees frequently mentioned that microbiologists or other laboratory technicians would volunteer their time to learn and transition to bioinformatic roles, but that the technical training necessary to build pipelines remained a significant hurdle. To help facilitate the standardized use of reference pipelines, public health agencies need a software ecosystem that readily provides access to easily deployed pipelines. To ensure that limited informatic training does not act as a barrier to use, reference pipelines should be containerized for easy deployment to a range of environments, wrapped to allow interaction via graphical user interfaces, and accessible via web-based entry. Additionally, pipelines should be completely open-source, make use of common, non-proprietary file formats, and be developed transparently in an environment that supports feedback and issue tracking. In our experience, many open-source projects and software are critically important, yet under-funded and developers over-taxed. We emphasize that this type of development effort will need to be appropriately incentivized, and will require large funding sources to support initial and sustained development and maintenance.

What might this deployment platform look like? We envisage a model where all reference pipelines and validation datasets are catalogued in a registry. Registry entries should provide information about the pipeline, such as what it does, what inputs it takes and outputs it provides, as well as information about where the pipeline is hosted and how to access it. These pipelines should be containerized so that individuals familiar with command line interfaces can source and run a pipeline from a container hub (e.g. Docker, Singularity) using minimal shell scripts or pull and run commands. In addition, an open-source initiative could be funded to write and maintain graphical user interfaces for interacting with these same pipelines. Having graphical user interfaces will ensure that individuals with less familiarity with command line interfaces and programming can still access and use bioinformatic tools. Graphical user interfaces could either wrap the sourcing and running process, or wrap pipelines that are hosted directly on a specific public health server. These options are not mutually exclusive, and we imagine that both a broad registry and a more narrow shared computational service that hosts the most frequently used pipelines would co-exist. We emphasize that both registries and deployment platforms for shared computational services must host multiple generations of pipelines, clearly indicating which versions are vetted reference pipelines, which pipelines are beta tests of future generations, and which pipelines are deprecated. Having this versioning will allow pipeline improvement and development while maintaining access to the hardened reference pipelines.

The pipeline deployment platform should be developed as an open-source community project. Specific instances of the platform could be deployed on distinct compute infrastructures managed by various public health agencies or networks of agencies. The compute infrastructure could be cloud-based (see Proposal 7) or could use an in-house cluster. Creating and managing instances of the platform could be performed either by in-house computational personnel or could be performed as Software as a Service (SaaS), where SaaS is provided by a non-profit or a company that charges individual users for their compute. Notably, open-source platforms with SaaS options have been successful, as in the case of Arvados.


Package & Pipeline Registry
Package & Pipeline Hub
mafft
raxml
snippy
flu-whd
campy-wgmlst
GUI Access Portal
Call haploid variants with snippy
Reference-grade wgMLST for Campylobacter

```
Package: mafft
Version: 1.4.3
Install: cd mafft-x.x/core/
         make clean
         make
         su
         make install

Package: RAxML
Version: 8.2.12
Install: conda install -c bioconda raxml

Package: snippy
Version: 4.3.6
Install: brew install brewsci/bio/snippy

Pipeline: flu_within_host_diversity
Grade: Research
Validation: Unvalidated
Generation: 2.2
Source: docker pull flu-whd

Pipeline: campy_wgMLST
Grade: Reference
Validation: CLIA-validated
Generation: 1.4
Source: docker pull campy-wgmlst
```

Figure 2. Three important components of this ecosystem: a package and pipeline registry, a hub where containerized packages and pipelines are hosted, and a graphical user interface access portal for running pipelines and packages.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 January 2020 doi:10.20944/preprints202001.0107.v1

The creation and maintenance of a bioinformatics infrastructure for the public health community will require ongoing social, political, and technical investment. In developing said platform, many different considerations must be tackled. These include, but are not limited to:

(1) developing and/or implementing validation criteria for bioinformatic analyses,

(2) developing new bioinformatics tools and pipelines,

(3) extending tools that already exist such that they work in the deployment platform environment,

(4) determining who will host pipelines and graphical interface portals to pipelines, and

(5) communicating the requirements of the community to developers and users. These requirements would include, for example, the data models, API standards, and pipeline documentation standards.

Governance of the deployment platform should focus on engaging with developers and users. This effort is necessary to ensure that development effort is sustainably supported, and to evaluate the effectiveness of the platform in providing open-access, low-barrier bioinformatics and genomic analyses.

5. Develop and support pipelines for data visualization, exploration, and automated analysis

Almost uniformly, interviewees wanted access to computational tools to visualize and explore genomic data. Their goals were frequently to understand their data better, to improve communication of genomic findings to surveillance epidemiologists, and to automate routine analyses and generation of reports. Currently, genomic visualizations such as phylogenetic trees, minimum spanning trees, SNP matrices etc., are often shared manually, by sending images over email. Surveillance epidemiologists may then manually add in epidemiologic data to look at how exposures or other relevant information correlate with the genomic data. This process is clunky and inefficient, and public health agencies would benefit from tools for automated data joining.

A variety of software for exploring and visualizing genomic analyses already exist. These include platforms such as Microreact, Nextstrain, and PHYLOViZ (discussed in more detail in the software section). Despite their value for understanding genomic data, these tools are variably implemented in public health settings. We found that the primary impediments to deploying this software were insufficient technical knowledge to support an instance of the software, and the separation of epidemiologic and genomic data streams, which is done for security purposes. The lack of automated methods for rejoining genomic and epidemiologic data after bioinformatic processing limits the information that can be visualized on a phylogeny or other genomic data object, thereby reducing the utility of the visualizations. Development of data joining tools has been hampered at least in part due to the form of epidemiologic data, which may be variably complete and is often non-standardized, especially when collected during rapidly evolving outbreak investigations and response activities.

To ameliorate this situation, analytic and visualization pipelines ought to be separated from assembly pipelines. We emphasize this point because, while this division is relatively common in academic settings, the majority of bioinformatic pipelines we saw in public health programs ran from raw reads through to a genomic visualization in a monolithic series of computations. Taking a more functional, microservices-oriented approach to bioinformatic software development will likely improve the flexibility, performance, scalability, and maintainability of public health bioinformatic applications and pipelines. Separating the assembly and analytic processes should also help overcome some of the issues of data classification and server security. Genomes can be assembled in the absence of epidemiologic data, and this can occur on lower security, scientific computing infrastructures. Then, subsets of epidemiologic data housed on secure servers can be joined with the assembled genomes, and the joined data objects can be run through visualization pipelines. Since the data should be sourced via API calls, different levels of security authorization can be required to source different components of the epidemiologic data with appropriate encryption and access controls. Analytical pipelines would export information for visualization in interactive browser-based portals. If high security is necessary, these could be served locally, and if not, they could be shared more widely over the web.

Notably, much of the infrastructure and code base for visualization pipelines has already been developed; the challenge has been how to integrate and run these pipelines within public health agencies. To support this effort we need to: (1) standardize the structure of genomic data through the use of data models, (2) standardize epidemiologic data via adoption of ontologies, and (3) build the API infrastructure to source and join data streams while respecting security considerations. Again, we stress that the additional development effort needed to adapt current tools to work within a new software ecosystem, and to support them over time, will require sustained funding mechanisms.

6. Improve reproducibility of bioinformatic analyses

In Proposal 4 we described a deployment platform for public health bioinformatics. Here, we describe in greater detail how individual pipelines should be developed to ensure that they are highly reproducible. Consistent with common practice in academia, we found that many public health programs use similar, but distinct, pipelines for bioinformatic analysis; the wheel is frequently reinvented. While these pipelines all use a relatively narrow suite of the same open-source software programs, the lack of standardization across bioinformatic pipelines impacts the comparability of data and results across agencies. This lack of comparability was a major concern that was voiced frequently. The need for standardization and highly reproducible assays is particularly acute in public health. In contrast to academic settings, sequencing assays in public health need to be sufficiently robust and reproducible to meet government regulated standards.

In addition to stable reference pipelines, interviewees also expressed a need to use custom pipelines. Often this need was brought up by frontline health departments investigating questions of local public health importance. We imagine that there may in fact be a high degree of overlap in the types of questions an agency will investigate at the local level. Therefore, supporting wider access to non-reference pipelines may actually help to harmonize these analyses as well.

To have a high degree of reproducibility, agencies need to be able to use the same pipelines as each other, and the pipelines and their component software packages must be stably deployed. We discuss below possible strategies for supporting reproducibility. In particular, we consider how pipeline development can prioritize reproducibility, how the process of running pipelines can be made more reproducible via containerization and workflow management, and the need for rigorous auditing and validation to verify reproducibility.

Versioning.

When considering reproducibility, we must consider three aspects of pipelines: the data being assembled, individual programs being piped together, and the packaged pipeline as a whole. All reference datasets, component software programs, and whole pipelines should be version controlled. Versioning software components, in addition to full pipelines, allows all changes to be tracked and documented, and ensures that developers can roll-back undesired changes. Versioned pipelines can then also be cloned, allowing development of newer generations of the pipeline or offshoot pipelines without inhibiting access to the stable reference analysis. Facilitating the coexistence of reference and customized pipelines together is critical to widespread adoption of a single bioinformatic pipeline deployment platform.

Containerization.

Similar to versioning, developers should containerize pipelines and any software packages that are used outside of workflows (nested containers are not well supported at this time). We propose using containerized pipelines for the following reasons:

(1) Containerization increases the reproducibility of analyses because you can run the same pipeline in the exact same computing environment as someone else. This consistency in the compute environment limits issues where missing dependencies, or differences in versioning of dependencies, change the way a pipeline runs.

(2) Containerized pipelines are shareable. Facilitating pipeline sharing enables agencies to run the exact same pipeline as each other, which is preferable to each agency attempting to build a similar pipeline from documentation.

(3) Containerization promotes software stability and reproducibility because a program can maintain replicate instances of a workflow. Being able to host old pipelines alongside new ones under development ensures that access to reference pipelines is maintained. Having replicate instances also allows developers to benchmark pipelines side-by-side. This ability to compare workflows systematically within the same environment is critical to ensuring that bioinformatic assays remain valid even as they are updated.

(4) Containerized pipelines can be run using automated workflow managers, a quality that improves reproducibility.

(5) Containerization reduces the burden of reproducibility; while someone must verify that they are using the correct version of a container, they do not then need to check the versioning of all component software and dependencies.

Containerizing software programs could be started fresh within public health, or could make use of other containerization projects, such as BioContainers, BioBoxes, or FlowCraft. Bioinformaticians within public health have also begun to containerize useful software, such as the library of docker builds maintained by the State Public Health Bioinformatics group (StaPH-B) (github.com/StaPH-B/docker-builds).

Containerized reference pipelines should be released as versioned generations. All generations of a pipeline should be concurrently hosted on the platform to ensure historical compatibility of bioinformatic analyses. New generations of pipelines should be thoroughly validated and released according to a stable release cycle. Additionally, beta versions should be released to the portal regularly to allow user testing and commenting.

Auditability.
Within a public health context, all transformations of data should be describable and recorded. Pipelines and workflows should create auditable reports that include the name and version of the program running, as well as which input parameters were used, especially if these vary from default. Pipeline runs should automatically store intermediate files in standardized formats. Having access to intermediate files is important as they can reveal the presence of discrepancies, and where they were introduced, within an analysis. Additionally, intermediate files help describe how data were transformed during the analysis.

Validation.
Despite previous work to develop validation datasets (described in [8]), many interviewees mentioned that they would benefit from further development of structured validation criteria for bioinformatic assembly pipelines. We suggest that agencies perform end-to-end proficiency testing of WGS protocols, including both the laboratory and bioinformatic portions of the assay. We imagine that agencies at higher levels of jurisdictional authority would be responsible for developing the validation metrics, given the need for these standards to be unified across all levels of public health. Finally, the bioinformatics deployment platform should clearly communicate which pipelines, at which generation, have been formally validated.

Workflow management.
One of best the strategies for writing reproducible and auditable pipelines is to design them as automated workflows. While pipelines can be written as single scripts, specifying pipelines in workflow languages creates self-documenting pipelines. To maximize portability, workflows could be written in Common Workflow Language (CWL), which would allow them to run on various deployment platforms such as Arvados, Terra(FireCloud), and eventually also on Galaxy. Specifying workflows in CWL also allows users to automatedly translate their workflows into other workflow languages. Alternatively, pipelines could be written with other workflow systems, such as Snakemake or Nextflow. While potentially not as portable, these workflow systems have high uptake in biology, and may be more familiar to developers in public health.

7. Use cloud computing to improve the scalability and accessibility of bioinformatic analysis

To date, the adoption of cloud computing in public health has been hindered by issues with process, compliance, and acquisition of cloud services by governmental agencies at all levels of jurisdictional authority. However, as cloud services become increasingly feasible for government agencies to access, we expect the utility of these resources to increase. The ability to access cloud services will allow smaller jurisdictions, or those with limited infrastructure and resources, to support sophisticated bioinformatics capabilities without incurring significant capital or operational expenditures. This will be an important leveler for low and middle income countries, who can take advantage of a community-driven ecosystem of deployable and scalable bioinformatic tools and workflows.

Supporting accessibility.
Within the United States, adoption of BioNumerics gave frontline public health agencies greater autonomy to investigate diseases that were a priority at the local level. As we transition to an open-source software ecosystem, this autonomy needs to be maintained. Access to cloud-based bioinformatic analyses puts these tools at the frontline of public health, helping smaller public health agencies do more sophisticated analyses, and reducing response lags in outbreak scenarios. Using the cloud supports broad decentralized access to bioinformatics by ensuring that reference data, testing datasets, and pipelines are available from one place.

Using a cloud-based bioinformatic platform would also increase accessibility by decreasing economic burden on small or lower-resource labs. If using a cloud-based bioinformatic ecosystem, only one or a few high performance computing environments need to be managed. Thus not every institution needs to pay for server hardware or the highly remunerated workforce necessary to maintain a cluster, although they might have to pay for their usage of the cloud-based ecosystem unless centrally funded. Shifting to cloud computing converts capital expenses to operational expenses, which hopefully will make it easier to spend money on compute resources. While many agencies likely understand the traditional capital and operational expenditures associated with purchasing and maintaining servers, probably fewer currently have a good understanding of how cloud computing operational expenditures compound, and how to install necessary controls on them. Thus to ease expenditure concerns and smooth adoption, we propose that public health programs receive training on how cloud operational expenditures work, how to install controls, and how to train users who are purchasing resources.

Increasing capacity and efficiency.
Movement to a cloud-based analytic ecosystem allows dynamic scaling of compute resources, which allows the ecosystem to adapt to changing data storage needs, changing amounts of sequence analysis, and changes in the compute intensity of projects. While we absolutely face a growing need for compute power, that rise will have spikes and drops along the way. Cloud-based ecosystems can be designed to handle this variation elegantly, reducing the risk of compute underutilization or insufficient access to resources.

Cloud-based software ecosystems have been successfully used in academic settings to provide small labs with access to high performance computing and software, and developers could look at platforms such as the National Science Foundation supported XSEDE ecosystem as an example. That said, we recognize that there will be challenges to deploying cloud-based analyses for all possible public health scenarios. We imagine that the greatest challenges to overcome will be connectivity issues in lower resource settings and the need to update agency-specific data and patient privacy policies to permit cloud-based computing. Overcoming these hurdles will require open communication between public health agencies and cloud computing providers, such that cloud services can be tailored to the needs and standards of public health agencies.

8. Support new infrastructure and software development demands with technical personnel

Adopting the proposals outlined above will require the support of a more technical, computationally-oriented workforce. In addition to bioinformaticians and genomic epidemiologists, supporting this infrastructure will require personnel with expertise in high-performance computing, cloud computing systems engineering, network/storage engineering, data science, and software development. Almost every program we interviewed mentioned that attracting and retaining this workforce was challenging for a number of reasons. Lower compensation, lack of access to newer technologies, and the frustrations of working within a government agency all affect workforce development.

What often does attract bioinformaticians and software developers to working in public health is the opportunity to have impact. The meaningfulness of work in public health and the ability to use one's expertise to tangibly improve people's lives are opportunities that private sector positions generally do not offer. Thus, in developing their technical workforce, public health agencies should highlight these factors when recruiting. Additionally, laboratory microbiologists are pivoting towards more bioinformatics-heavy roles, often by learning these new skills on their own. In addition to improving recruitment, public health agencies should support training programs that facilitate the transition from bench microbiology to bioinformatics. Due to their prior role, these individuals have an incredible wealth of knowledge about upstream sequencing process that can aid in troubleshooting and evaluating bioinformatic processes.

We found that successful recruitment of technical personnel into public health often occurred through connections with academic institutions. Generally, these relationships were forged when public health agencies reached out to access high-performance computing, or when they sought graduate-level students to work on informatic questions for thesis or practicum projects. Additionally, technical personnel can be recruited through fellowship programs. Successful examples include the CDC Public Health Informatics Fellowship Program and the APHL-CDC Bioinformatics Fellowship Program.

Recruitment is just one piece of workforce development, and interviewees mentioned that they also found it challenging to retain technical staff. Likely because bioinformatics and software engineering are new careers within public health, mechanisms for career advancement are not well developed at this time. We found that some agencies did not have formal job descriptions specific to computational disciplines, let alone competency and assessment criteria, or mechanisms to move into leadership roles. To sustain a computational workforce, public health agencies should create clear descriptions of the disciplines and job series for bioinformaticians, data scientists, and software engineers. We imagine that describing these positions and developing career trajectories will make public health a more attractive career option.

9. Improve the integration of genomic epidemiology with traditional epidemiology

From discussions with interviewees, we found that the degree of integration of genomic and traditional epidemiology varied highly across programs. Some divisions and institutions had open collaborations between bioinformaticians and epidemiologists, often with weekly meetings to discuss ongoing outbreaks or future surveillance directions. In other cases, we found that bioinformaticians had limited contact with epidemiologists; bioinformaticians would send out routine reports, but little information about how the genomic data were interpreted, or questions requiring follow-up, were communicated back. While genomic and traditional epidemiology work synergistically, the training required to understand and analyze genomic data versus case data is distinct. Many individuals working in public health do not have both types of training. Thus, facilitating open communication and collaboration between bioinformaticians and epidemiologists is fundamental to supporting integrated surveillance systems.

We believe that integration of these domains can be improved by providing basic training in analyzing both types of data. Many of the bioinformaticians we spoke with said that they would appreciate having a better understanding of traditional epidemiologic methods. We interacted with fewer epidemiologists during our interviews, however those we spoke with commented that they found interpretation of genomic data challenging without specific training. Potential ways to provide this training include online and in-person courses. Interviewees mentioned that sustained courses, that met once a week over a longer period of time, were more helpful as this schedule allowed time to practice newly acquired skills and ask follow-up questions. In addition to providing further training, it might be helpful to have a team of genomic epidemiologists and bioinformaticians available to work with different agencies temporarily, providing training and support to onboard new techniques and systems for genomic epidemiologic analysis.

Integrating genomic and traditional epidemiology will require technical developments in addition to the social ones discussed above. From a technical standpoint, integration of genomic and epidemiologic data will require more sophisticated databasing approaches, included programmatic data sourcing and merging that respects security levels, use of ontologies to standardize data reporting formats for both surveillance data and genomic data, and machine-learning methods for data classification, tagging, and cleaning. The need to unify and harmonize systems to improve genomic epidemiology has the added benefit of developing a database infrastructure that could facilitate cross-agency data sharing. During the course of our interviews, interviewees frequently mentioned the need to see beyond human surveillance data in order to fully comprehend the source and extent of an outbreak. Ideally, the use of standardized data models and ontologies, as well platforms accessible via cloud computing, would unify analytic platforms and standards in multiple agencies. While cross-agency integration would be politically challenging, it would exponentially improve public health programs' ability to perform surveillance within a One Health paradigm.

doi:10.20944/preprints202001.0107.v1

10. Develop best practices to support open data sharing

In an interconnected world where disease transmission occurs across borders, environments, and species, the best surveillance system would support data sharing across institutions and agencies, both within country and between countries. Every agency we interviewed understood the value of data sharing, and many had anecdotes where their inability to share or receive data hampered surveillance activities. Despite the recognized value of sharing data, putting it into practice is challenging. In part, this is due to the critical need to protect patient privacy. Public health programs rightfully must follow rules that govern how personally-identifiable information (PII) is shared. However, in practice these rules can make data sharing convoluted, since definitions of PII vary by disease incidence and geography, and different places have different laws and protections governing the use, storage, and transmission of PII. In order to develop a data sharing system that functions well for public health, we think that data sharing needs to:

(1) be easy to do, so that it is not a burden,

(2) occur along trusted channels, and

(3) be granular, so that access to different levels of data can be filtered based on security and legal constraints.

Large, diverse genomic datasets from many groups are greater than the sum of their parts, and their utility has built momentum for greater data sharing within some sectors of public health. Perhaps the best example of this is PulseNet, a large, multi-agency network that supports within-network data sharing. In line with our proposals, PulseNet data sharing occurs along trusted channels, built on memorandums of understanding with each of the collaborating partners. These memorandums describe how data will be shared, with whom, and at what granularity, ensuring compliance with state and federal law. Importantly, the PulseNet sharing system is also relatively easy to use, and is integrated into BioNumerics. Detailed and complete data about a sample can be added to local BioNumerics databases, and subsets of that data can be shared with PulseNet via easy interaction with the BioNumerics graphical user interface.

While PulseNet's efforts have pushed data sharing forward immensely, we still find instances where released metadata are too coarse to provide meaningful genomic epidemiologic inference. To make improvements, we encourage a deeper conversation about identifiability, including what the risks are and what information we might lose by masking data. This discussion could be initiated by considering the following questions, adapted from Cologne et al [9]. What is the probability that someone could identify a case given the released metadata? What are the consequences of identification, should it occur? How might masking or omitting metadata affect their analytical utility? The answers to these questions will vary by pathogen, by disease incidence, by geography, and by host. As such, maintaining the integrity of data sharing will require frequent re-evaluation of the risk of identifiability, consequences of identification, and assessment of how analytical utility may be lost when masking data.

A secondary concern about data release is scooping, a process in which another group analyzes and/or publishes on data without permission of the data generators. This is a consistent concern with data sharing also found outside of public health within academia. While this concern is hard to allay, in our experience scooping is more rare that one imagines it will be. Additionally, as agencies develop their capacity to perform thorough genomic epidemiologic analysis quickly, data analysis will occur on roughly the same time scale as data release, which should also reduce the risk of getting scooped.

We emphasize that the development of increased data openness in public health cannot be all or nothing; if it is, we will simply end up with a system where sharing is limited. Instead, we should identify consistent small steps that can be taken to improve the openness of data, with the hope that open data and integrated databases improve surveillance and outbreak response sufficiently to warrant their continued development and maintenance.

Current software platforms and programs

Whole genome sequencing (WGS) is now a routine component of molecular biology, and there are a wide variety of applications that transform, analyze, and visualize WGS data. However, many of these tools run from UNIX-based command line interfaces that require technical knowledge to use. Additionally, some bioinformatic processes may be computationally intensive, requiring access to high-performance computing and knowledge of how to use cluster or cloud-based server infrastructure. This creates a mismatch between the currently available infrastructure and workforce within public health and what would be ideal to support large pathogen genomics programs. As the frontline for outbreak response and surveillance, public health institutions have broad mandates for WGS, usually requiring large amounts of sequencing and analysis of many different pathogens. Despite this, many agencies currently have minimal access to expertise in high-performance computing and bioinformatics. This means that an ideal informatic ecosystem for public health would be sufficiently accessible and intuitive to use that individuals would need only minimal formal training to analyze and visualize genomic data.

Compared to the large numbers of individual bioinformatic applications, there are fewer platforms that manage WGS data storage, host full pipelines for bioinformatic assembly, and provide visualization of the assemblies in a unified manner. We feel that the primary software hindrance to pathogen genomics in public health is not necessarily lack of access to bioinformatic tools, although this is certainly an additional problem in institutions where employees cannot access off-network computers. Rather, the greater need is to develop systems that perform sample management, automated storage and sharing, and host reproducible pipelines integrated with visualization and results sharing. By discussing what an ideal ecosystem might look like, we hope to improve the interoperability and usefulness of the platforms and software that already exist. We describe below some of the current platforms and tools that public health agencies regularly use.

doi:10.20944/preprints202001.0107.v1

Unified platforms for databasing and workflow management

BioNumerics.
Developed by Applied Maths (now bioMérieux) for standardized gel analysis in the era of pulsed field gel electrophoresis (PFGE), BioNumerics now also supports WGS analysis and is widely used in public health. BioNumerics provides a graphical user interface with access to multiple pipelines for genome assembly and analysis, including calling allele profiles and making minimum spanning trees. While interviewees appreciated the graphical user interface, they consistently mentioned that the greatest value of the software is the way it organizes and performs databasing, providing efficient sample management, archiving, and data sharing interfaces that are integrated with the bioinformatic pipelines. BioNumerics supports sourcing raw data from NCBI, pushing data to NCBI, and automated storage of various 'experiment types' (e.g. allelic profiles, SNP matrices) as these results are generated, in an easily searchable backend database. BioNumerics can run locally or make use of cluster and cloud-based servers. This means that users can access high-performance computing or analyze data that can't be shared on their local system. Within the United States, this platform has served to create a distributed network where state and local-level agencies have relatively high autonomy, something that any new software ecosystem must also support.

While BioNumerics has helped to bridge the transition from PFGE to WGS, there are a variety of limitations to the software. Most critically, BioNumerics is not open-source, and licensure costs can be prohibitively expensive for small institutions and institutions in low and middle income countries. Additionally, pipelines in BioNumerics generally run straight through from raw data to a visualization endpoint. While this type of pipeline is easy to use, the lack of modularity limits the types of comparative genomic analyses that can be performed, and impacts the users ability to flexibly change, and interact with, the genomic visualizations.

Integrated Rapid Infectious Disease Analysis (IRIDA).
Developed in Canada as a collaborative effort between the Public Health Agency of Canada, provincial public health agencies, and Canadian academic partners, IRIDA is a free, fully open-source software ecosystem for performing pathogen genomic analysis. The IRIDA platform performs data management and storage, and provides graphical user interfaces for bioinformatic pipelines executed by Galaxy. This provides users both the accessibility of an end-to-end workflow, and the flexibility to adapt or extend pipelines as desired. In addition to pipelines for bioinformatic assembly, IRIDA also supports integrated analytic pipelines that perform sequence typing, antimicrobial resistance prediction, and phylogenetic and phylogeographic reconstruction. Results and visualizations can be stored in IRIDA or transferred out to other applications and databases via a REST API. While IRIDA encourages and facilitates data sharing, independent instances of IRIDA can be run locally if data cannot be shared outside of an agency. Uniquely, IRIDA has a large emphasis on data standards, and will support the use of controlled vocabularies (ontologies) for genomic data and epidemiologic metadata, with the aim of standardizing data reporting and facilitating inter-agency data harmonization.

IRIDA has many of the qualities that we would seek to have in a wider public health software ecosystem; indeed it was built specifically to fill this gap in Canada. However, one of the major challenges facing IRIDA, and one that other public health programs will have to consider as well, is the support model. Despite allowing decentralized genomic analysis, IRIDA is almost

doi:10.20944/preprints202001.0107.v1

entirely centrally supported. Canada's reference laboratory, the National Microbiology Laboratory, provides the high-performance computing infrastructure for all provincial laboratories that use IRIDA, and is the only agency that can fully support software hosting and development in house. This centralized support model works to provide software and analytic infrastructure, but limits the degree of community support to continue growing the platform. This places a large amount of responsibility and ownership on a single group. Ideally, to transition this platform to a community-supported model, deployment of the ecosystem would be accompanied by targeted development of in-house capabilities at the regional laboratory level.

INNUENDO.

INNUENDO was developed by a large number of European partners with the goal of creating a unified platform for foodborne bacterial genomic assembly and analysis. The consortium brings together academic and governmental agencies from Finland, Estonia, Latvia, Portugal, Austria, and the Basque Autonomous Community in Spain. They also come from various sectors, including food safety, animal health, and human health. Similarly to IRIDA, INNUENDO is a fully open-source software platform that hosts pipelines for bioinformatic assembly and genomic analysis that are accessible via graphical user interfaces. INNUENDO also supports browser-based visualization, using a REST API for data transfer. INNUENDO uses pipelines built with flowcraft, an open-source application that allows easy, modular assembly of bioinformatic pipelines. Within this framework, each bioinformatic application called as part of the pipeline is containerized in Docker, ensuring greater reproducibility. These predefined workflows are also highly auditable, creating run reports with versioning and command information. Similarly to IRIDA, INNUENDO has many of the qualities that we seek from a bioinformatic ecosystem for public health, and it should be looked at closely as a model.

PathogenWatch.

PathogenWatch is an easy-to-use platform for genomic surveillance, consisting of a datastore for genomic assemblies and metadata and a web client for processing and visualisation. Broadly focused on contextualising genomes within larger datasets and rapidly delivering results, PathogenWatch performs analytics on genomic assemblies, firstly identifying species, then performing species-specific analyses, such as multi-locus sequence typing (MLST), core genome MLST, the prediction of antimicrobial resistance profiles, and the inference of phylogenetic trees. PathogenWatch also hosts curated pathogen-specific genomic datasets that users can explore and use to contextualize their own data. PathogenWatch utilises a dockerised plug-in architecture allowing the inclusion of additional informatic pipelines created by the community. PathogenWatch provides an integrated database and visualization system.

Workflow platforms

EDGE.

Developed at the Los Alamos National Laboratory, EDGE is a bioinformatic workflow platform that provides users with a web-portal to access bioinformatic pipelines via a graphical user interface. This platform makes genomic data processing and analysis accessible to users without extensive bioinformatic experience.

Application-specific platforms

IDseq.
IDseq is a centralized metagenomics platform that performs taxonomic identification of pathogens from uploaded FASTQ reads.

Mykrobe.
Mykrobe is an accessible, graphical user interface-based application that predicts the drug resistance profile of a pathogen given whole genome sequence information. Prediction is currently available for *Staphylococcus aureus* and *Mycobacterium tuberculosis.*

Visualization and data exploration software

Visualization platforms are already used to support public health surveillance. However, without computational expertise, it can be challenging to implement instances of the software. To make visualization software more accessible, these packages should be integrated into the greater bioinformatic ecosystem. For integration to work, we will need to have standardized assembly output formats and genomic data models that software applications can work with, as well as well-designed APIs to facilitate data transfer. Below we detail a subset of the genomic data visualization tools that are often used in public health.

Microreact.
Microreact is a flexible web application for linking and visualizing geographic, temporal, phylogenetic, network, and epidemiologic data. Microreact enables users to easily upload or link to files containing metadata, and/or tree and network files, to create a range of visualizations, including trees decorated with metadata, maps, timelines, and tables. Visualizations in Microreact are easy to share via permanent web links, and can be linked to within publications. Furthermore, a comprehensive API enables the extension of Microreact to dynamic data. The application is used extensively by the European CDC, the US CDC, and Public Health England.

Nextstrain.
The Nextstrain software suite performs inference and visualization of maximum likelihood and time-resolved phylogenetic trees. It has the capacity to annotate tips of the tree given sample information, and also reconstruct states at internal nodes in the tree. Additionally, Nextstrain allows visualization of alignment and sequence characteristics, such as reconstructing nucleotide and amino acid mutations along each branch in the tree. The Nextflu subsidiary, which infers changes in antibody cross-reactivity across the tree, is used by the World Health Organization to inform vaccine selection, and by the CDC Influenza Division to support surveillance.

PHYLOViZ.
Developed by the INNUENDO consortium, PHYLOViZ can infer allelic profiles and minimum spanning trees, and visualize associated metadata about pathogens on those trees. Additionally, the software will visualize distance matrices showing genetic distance between strains, and allows organization of that matrix by epidemiologic metadata. The software is accessible via a Java

desktop application or browser-based application.

MicrobeTrace.

MicrobeTrace is a web application-based inference and visualization tool for drawing contact networks and transmission graphs given epidemiologic and genomic data. It allows inference and exploration of an underlying network structure, and can annotate nodes and edges with epidemiologic data. MicrobeTrace is used frequently by CDC to support contact tracing efforts for HIV outbreaks. MicrobeTrace can also be used offline, which is an important quality for security considerations.

Our vision of a potential software ecosystem

Given our proposals, and the software tools that currently exist, what do we envisage an open-source software ecosystem in public health might look like? We imagine that the system would benefit from being highly modular, with genomic assembly and data processing separated from genomic analysis and visualization processes. Splitting these processes will maintain efficiency while allowing flexibility, enabling many different analyses to be performed without having to rerun assembly pipelines. Importantly, separating the assembly and the analytic processes also ensures that output from the assembly pipelines is archived, an important extension to current archival practices that focus primarily on storing raw sequencing reads. The primary pieces of this ecosystem would be databases, APIs, pipelines, and scripts that move data around.

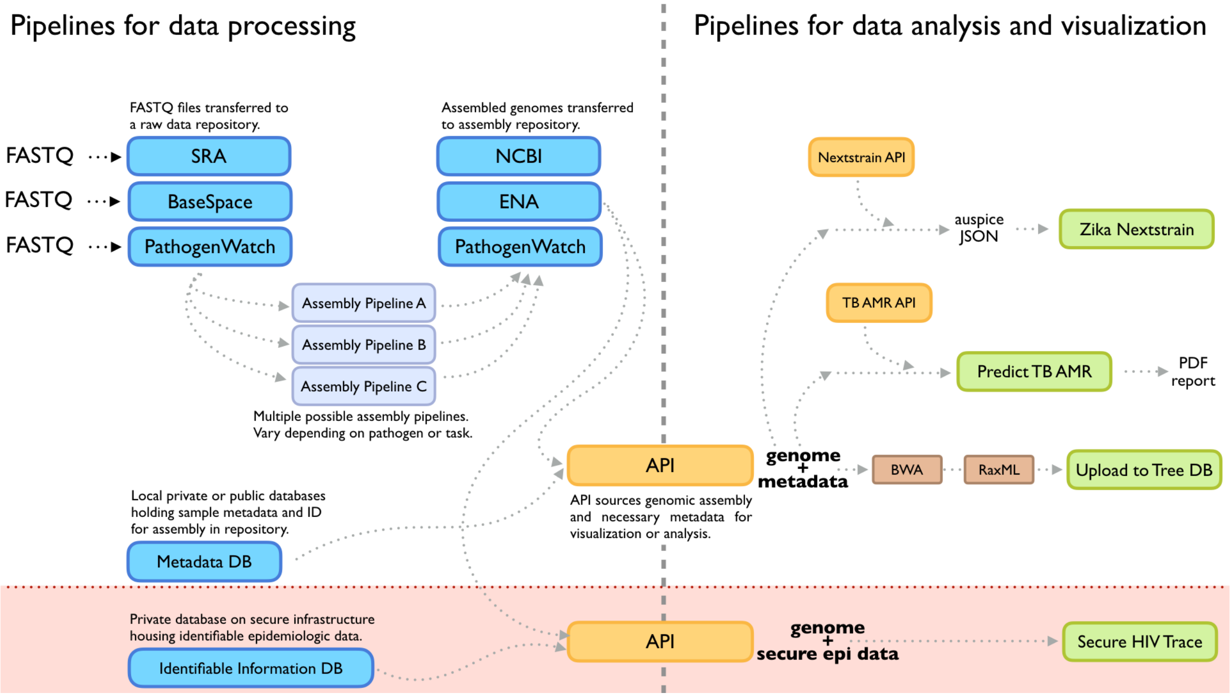

Pipelines for data processing
Pipelines for data analysis and visualization
FASTQ files transferred to a raw data repository.
Assembled genomes transferred to assembly repository.
FASTQ
FASTQ
FASTQ
SRA
BaseSpace
PathogenWatch
NCBI
ENA
PathogenWatch
Assembly Pipeline A
Assembly Pipeline B
Assembly Pipeline C
Multiple possible assembly pipelines. Vary depending on pathogen or task.
Local private or public databases holding sample metadata and ID for assembly in repository.
Metadata DB
API
API sources genomic assembly and necessary metadata for visualization or analysis.
Nextstrain API
auspice JSON
Zika Nextstrain
TB AMR API
Predict TB AMR
PDF report
genome + metadata
BWA
RaxML
Upload to Tree DB
Private database on secure infrastructure housing identifiable epidemiologic data.
Identifiable Information DB
API
genome + secure epi data
Secure HIV Trace

Figure 3. This schematic illustrates our vision of how an ecosystem in public health for bioinformatic assembly and genomic epidemiological analysis might look. We envisage a system where bioinformatic workflows are separated from genomic analysis and visualization workflows, with interaction and data sourcing mediated by APIs.

On the data assembly side, we consider three distinct types of databases: one for archiving or holding raw sequencing reads, one for archiving assembled data, and one for holding metadata about the samples. A variety of current databases could fill these positions, or the field could develop new databases if public health programs require additional utility. We imagine that the Sequence Read Archive would continue to serve as the primary raw reads database. But, if one has a metagenomic sample containing both pathogen and human reads, the reads could easily also be held in Illuminas BaseSpace platform instead. From here, raw reads could be assembled by one or more of the open-access pipelines; pipeline choice would be based around what type of assembly the user needs. A final portion of the pipeline should be the automatic depositing of the genomic assembly into the relevant database for that assembly type. This database could be a relevant NCBI database (e.g. NCBI Nucleotide, NCBI Pathogen Detection), any database that is part of the International Nucleotide Sequence Database Collaboration (e.g. DDBJ, ENA), or a pathogen specific assembly database (e.g. GISAID, ViPR). The critical component of this databasing is ensuring that the accession identifier is deposited into a third database, the metadata database. Within our design, the metadata database would be an in-house relational database that facilitates sample tracking and houses all relevant clinical and laboratory data about the sample according to a well-defined schema that can also accommodate long form entries. Likely, the metadata database would be better to license than to build, however we do not have a specific databasing platform that we suggest at this time. Importantly, metadata databases could also be secured, and house relevant personally identifiable patient information collected during epidemiologic investigations. Having these data live separately from the genomic data ensures that PII can be kept private when necessary. Data linking would occur via API calls; calls to the metadata database would pull relevant sample information and the assembly accession number, allowing the assembly to be sourced from the genomic database. Various metadata+genomic data combinations could be sourced depending on what data fields are necessary for the desired analytic or visualization pipeline.

Once genomic assemblies and relevant metadata have been combined, they would be piped to various analytic workflows, such as predicting antimicrobial resistance, making specific data structures such as phylogenetic trees, or preparing datasets or data objects for serving to interactive data visualization platforms. We imagine that there will be a wide array of different visualization and analytic pipelines in use; good APIs and complete, standardized metadata are necessary to support that breadth. Some of these analytic pipelines may be completely containerized end-to-end workflows that produce visualizations or reports. Others could make data objects, such as phylogenies, and submit these to a database for use in subsequent analyses, which could save compute time. Additionally, these pipelines could make API calls to external databases, such as antimicrobial resistance gene databases, thereby facilitating the integration of these new pipelines with tools and ecosystems that have already been built.

Conclusion

The shift toward extensive use of pathogen whole genome sequencing represents a turning point for public health agencies; agencies must pivot to accommodate a new data source that provides increased resolution for understanding disease dynamics, but that requires different tools and a changing workforce to support. While change poses challenges, we hope that these proposals provide direction that supports the public health community as it transitions.

Acknowledgements

We are grateful to all of the public health professionals who took the time to participate in these interviews and discuss their experiences with us in depth. Without them, this work would not have been possible. We thank all the participants in the Open Public Health Bioinformatics Meeting for contributing to this discussion, and in particular to David Aanensen, João Carriço, Richard Neher, and Daniel Park for their edits and suggestions to this paper. This work was supported by a grant from the Bill and Melinda Gates Foundation to the CDC Foundation. A.B. is supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1256082. T.B. is a Pew Biomedical Scholar and is supported by NIH R35 GM119774-01.

Disclaimer

The findings and conclusions in this report are those of the authors and do not necessarily represent the official position of the Centers for Diseases Control and Prevention.

Bibliography

1. Fielding RT, Taylor RN. Architectural styles and the design of network-based software architectures. vol. 7. University of California, Irvine Doctoral dissertation; 2000.

2. Hadfield J, Megill C, Bell SM, Huddleston J, Potter B, Callender C, et al. Nextstrain: real-time tracking of pathogen evolution. Bioinformatics. 2018;34(23):4121–4123.

3. Dooley DM, Griffiths EJ, Gosal GS, Buttigieg PL, Hoehndorf R, Lange MC, et al. FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. npj Science of Food. 2018;2(1):23.

4. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Scientific data. 2016;3.

5. Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, Teal TK. Good enough practices in scientific computing. PLoS Computational Biology. 2017;13(6).

6. Goodman A, Pepe A, Blocker AW, Borgman CL, Cranmer K, Crosas M, et al. Ten simple rules for the care and feeding of scientific data. PLoS Computational Biology. 2014;10(4).

7. Hart EM, Barmby P, LeBauer D, Michonneau F, Mount S, Mulrooney P, et al. Ten simple rules for digital data storage. PLoS Computational Biology. 2016;12(10).

8. Timme RE, Rand H, Shumway M, Trees EK, Simmons M, Agarwala R, et al. Benchmark datasets for phylogenomic pipeline validation, applications for foodborne pathogen surveillance. PeerJ. 2017;5:e3893.

9. Cologne J, Grant EJ, Nakashima E, Chen Y, Funamoto S, Katayama H. Protecting Privacy of Shared Epidemiologic Data without Compromising Analysis Potential. Journal of environmental and public health. 2012;2012.