MDPI

Article

NCBI's Virus Discovery Hackathon: Engaging Research Communities to Identify Cloud Infrastructure Requirements

Firstname Lastname [1,†,‡], Firstname Lastname [1,‡] and Firstname Lastname [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)
† Current address: Affiliation 3
‡ These authors contributed equally to this work.

Version July 17, 2019 submitted to Genes

Abstract: A wealth of viral data sits untapped in publicly-available metagenomic datasets when it might be extracted to create a usable index for the virological research community. We hypothesized that work of this complexity and scale could be done in a hackathon setting. Ten teams comprised of over 40 participants from 6 countries, assembled to create a crowdsourced pipeline of datasets in a three-day event on the San Diego State University campus starting January 9th, 2019. Contiguous assemblies (contigs) were pre-assembled by National Center for Biotechnology Information (NCBI) staff using the SKESA algorithm for a large sample of 141,000 metagenomic datasets from the NCBI Sequence Read Archive (SRA). During the Hackathon contigs were aligned using BLAST against all known virus genomes, labeled with domains, clustered, annotated, and assigned metadata. The work yielded valuable insights into both SRA data and the cloud infrastructure required to support such efforts, and the scientific findings will be extended during a follow-up event.

Keywords: keyword 1; keyword 2; keyword 3 (list three to ten pertinent keywords specific to the article, yet reasonably common within the subject discipline.)

1. Introduction

While advances in sequencing technology have greatly reduced the cost of whole genome sequencing [1], it has given rise to new problems, especially related to data analysis and management. As the number of bases in the Sequence Read Archive [2] exceeds 33 petabases (June 2019), the difficulty to navigate and analyze all of this data has grown as well. Furthermore, as the number of data warehouses grows with the increased accessibility of the technology, the need to support interoperability of data types increases. To address these issues, the National Institutes of Health (NIH) launched the Science and Technology Research Infrastructure for Discovery, Experimentation, and Sustainability (STRIDES, [3]) initiative. "Through the STRIDES Initiative partnerships, NIH will test and assess models of cloud infrastructure for NIH-funded datasets and repositories."

As part of the initiative, the National Center for Biotechnology Information (NCBI) [4] launched a series of hackathons (see biohackathons.github.io), the first of which, the Virus Discovery hackathon, was held in January 2019 in San Diego. These events gather researchers for three days to work on projects around a topic, and provide an opportunity to quickly proptype a solution or a set of tools to address a community need. NCBI hackathons also facilitate networking among researchers, and allow NCBI staff to identify opportunities to improve their services. While previous hackathons were not specifically focused on working with large volumes of data or compute intensive tasks in the cloud, they provide a framework with which to engage the research community. As part of the STRIDES Initiative the hackathons are particularly focused on allowing researchers to work with large amounts of data in a cloud environment, in an effort to identify the needs and challenges of this new research environment. Typically topics are developed in conjunction with a host researcher and assigned to different working groups. Team leaders of these working groups are consulted to further refine the topics after some initial recruitment. On the first day of the hackathon, the teams further scope directions, and then continue to iterate on development goals over the course of the three days. Additionally, one writer from each group participates in a break-out session each day to help guide documentation of the work done.

While there are a number of commercial cloud providers, including Amazon (https://aws.amazon.com), Microsoft (https://azure.microsoft.com), and Google (https://cloud.google.com), at the time of this hackathon an agreement had been reached with Google as part of STRIDES. Google's cloud platform offers scalable nodes, with highly configurable access-control settings as well as an SQL-like database infrastructure, BigQuery. Details on cloud infrastructure can be reviewed here. Briefly, cloud compute refers to remotely hosted computers, some parts of which can dynamically access a single compute instance. This access to computing allows research scientists and organizations to use large and scalable computing resources without investing in the required infrastructure and maintenance. While this lowers the barrier to access supercomputer-type resources, it does not provide a comprehensive solution to the general scientific public. The main barriers to adoption by researchers include 1) modest experience in UNIX command-line type environments, and 2) the ineffectiveness of most commonly used bioinformatics tools to leverage the compute resources available in a cloud-computing setting. STRIDES hackathons aim to identify to what extent these barriers impact working researchers, and how they would like them addressed.

One of the fastest growing sources of public biological data is Next Generation Sequencing (NGS) data, housed in NCBI's Sequence Read Archive (SRA) [5]. SRA includes results from amplicon and whole genome shot-gun studies, conducted on a variety of sequencing platforms. The data is derived from research in many fields such as personalized medicine, disease diagnosis, viral or bacterial evolution, sequencing efforts targeting endangered animals, and sequencing of economically significant crops, among others. Despite the scientific potential of these datasets, there are several impediments to their usage. For one, sample metadata standards can vary between studies, making it difficult to identify datasets based on a common set of descriptive sample attribute terms. Moreover, while the content of some datasets is explicit, this is often not the case, particularly in those samples derived from complex mixtures of organisms like those from human gut and environmental samples. In these cases,

actual organismal content may not be known, either because it was never fully evaluated or because the content includes novel organisms not described in reference sets or otherwise undocumented, i.e. the so-called "dark matter" [6].

Understanding the microbial composition of different environments is necessary to support comparisons between samples and to establish relationships between genetics and biological phenomena. If such information was available for all SRA datasets, it would greatly improve both findability of specific datasets and the quality of analysis that could be conducted. However, determining the organismal content of a sample is not always an easy task as it typically requires comparisons to existing genome references. This can be difficult when samples include viruses because only a small portion of Earth's viral diversity has been identified and made available in reference sets [7]. Even when content can be identified, the very large size of SRA datasets present a significant scalability problem, and strategies must be developed to support large scale organismal content analysis in order to provide an index of this content for use in data search and retrieval. To that end, the first STRIDES hackathon engaged researchers working in this field to leverage the computational power of the Google cloud environment and test the applicability of several bioinformatic approaches to the identification of both known and novel viruses in existing, public SRA datasets

Here are presented the general results of these efforts, with an emphasis on challenges participants faced in conducting their work. Firstly, the scientific staff involved in the hackathon is presented, including their demographics and research backgrounds. Scientists were organized in teams which roughly corresponds to the different research sections found in this article, these include: data selection, taxonomic and cluster identification, domain and gene annotation.

Table 1. DUMMY: Participant dempgraphics.

Demodata 1	Demodata 2	Demodata 3
Data 1	data	data
entry 2	data	data

2. Results

2.1. Hackathon Planning and Preparation

The number of metagenomic datasets in the SRA database is steadily increasing, albeit not all the information that each SRA contains has been exploited to the fullest, e.g. not all species within sequencing datasets are routinely identified. A major hurdle for a detailed analysis of metagenomic datasets is the lack of readily available hardware and analysis pipelines. The goal of this hackathon was to identify user needs for standard NGS data analysis in a cloud environment as it can offer more computational power than is available on local processors. Viruses are present in virtually all organisms and environments, but only a limited number have been identified so far [8]. Therefore, virus sequences present a suitable model for NGS data mining.

Robert Edwards at San Diego State University agreed to sponsor the event and X participants registered for the event. Participant demographics are outlined in TABLE X. Participants came from a variety of academic backgrounds and countries, worked at a variety of institution types and at all stages of their career from training to senior investigator. The wide range of backgrounds allowed us to get a broad perspective on the hurdles faced by researchers working in a cloud environment.

After participants had been identified, teams were developed and team leaders identified. The team leaders were invited to an online, pre-hackathon event to orient them to the data and working in a cloud environment. This also allowed us to further refine the scope of the event and the focus for the various groups. At this event a data selection strategy was settled upon and a general approach was decided upon for most of the other groups (outlined in FIGX). Unlike in a typical NCBI hackathon, the groups were not working on independent projects, but were instead dependent on the work of "up-stream" groups. At the time of the hackathon, an agreement for data hosting had been reached only with Google, and so data was uploaded to the google cloud environment; all data uploaded was already publicly available via the NCBI's SRA resource (link). The data chosen to be uploaded for this event, along with the selection criteria, is described in the following section.

The pre-hackathon event highlighted the need for more of an introduction to doing bioinformatics in the google cloud environment, as well as an opportunity to improve workflows by pre-installing common tools on the VMs to be used by hackathoners, both of which were addressed before the actual event. The documentation developed can be found at the github repository linked above, though it will likely need to go through more revisions for it to adequately address the needs of researchers new to working in a cloud environment. The pre-installed tools are outlined in Supplemental Figure 1, and were chosen based on REF. Jupyter was found to be a popular environment to work from, and was not preinstalled. Work to identify the best Jupyter-style solution to hackathon needs is ongoing, and includes exploration of github's Binder, Google's CoLab, and custom Jupyter and JupyterHub set-ups. Having a dedicated IT support person on site was immensely helpful, for the various technical issues that are bound to arise at this type of event, but also to facilitate launching VMs as necessary for participants and adjusting hardware specs as necessary. Of note, it was found to be import to launch instances with a fixed external IP to prevent confusion when an instance is taken down and relaunched for any number of reasons.

Figure 1. DUMMY: Preinstalled tools.

Figure 2. DUMMY: Assembly strategy.

2.2. Data Selection

Whole Genome Sequences database (WGS) were targeted for inclusion, as we decided to focus on viruses from metagenomic studies for this event, and amplicon sequencing results do not represent this target. A total of 141,676 SRRs were processed using PARTIE [9] to determine how many of the datasets are whole genome sequences (WGS). Our results showed that 85,200 SRRs were WGS and these were analyzed further as part of the hackathon. However, during the hackathon, we first started with 2,953, a smaller test dataset consisting of samples that were randomly selected (1,000), selected based on the size of the dataset (999), and then based on size and the relatively large percentage of phage content (999). A complete list of SRR accession numbers that were part of each category can be found in our GitHub repository (https://github.com/NCBI-Hackathons/VirusDiscoveryProject/tree/master/DataSelection). The data selection pipeline is outlined in FIGX.

From this filtered data set, approximately 55 million contigs were assembled, though this represents only half of the raw reads. The participants were pleased with the contigs, and found them a useful way to get insight into a SRRs genomic content. That said, there was interest in exploring the suitability of different assemblers for this task. Given the heterogeneity of SRA data, preselecting the data was critical to the success of the event. However, given that the groups weren't working on independent projects and that they weren't all familiar with working with such a volume of data, the event might have been improved by identifying a much smaller subset for development and testing. Further, pre-selecting data-sets suitable for each group would have alleviated some of the issues associated with the groups being dependent on each other's work.

Figure 3. DUMMY: Pipeline.

2.3. Data Segmentation

As outlined in Figure 3, contigs were first pre-filtered based on size by removing all contigs shorter than 1 kb in length to increase data processing speed and leave only contigs more likely to provide meaningful hits. The remaining 4,223,563 contigs were then screened by BLASTn [10] against the virus RefSeq database [11] using a cut-off e-value of ≤ 0.001 and classified into three categories based on the average nucleotide identity (ANI) and alignment coverage (Figure 4):

Known-knowns: 12,650 contigs with high similarity to a known RefSeq virus genome, with ANI $< 85\%$ and contig coverage $> 80\%$. These contigs showed similarity to bacteriophage. In particular, 19

bacteriophage species showed hits to more than 100 contigs and, specifically, crAssphage comprising $\approx$ 27% of all known-known contigs.

Known-unknown: 6,549 contigs moderately similar to known viruses. This category was further divided into two subcategories. The first category contains 4,713 contigs with high similarity to known viruses (>85% ANI) but the alignment covers between 50-80% of the contig length. The second category contains 1,836 contigs with a lower alignment similarity to known viruses (50-85% ANI) and cover >50% of the contig length. These likely belong to either somewhat distant relatives of known viruses, close relatives that have undergone recombination other viruses, known viruses with structural variants, or prophage where the regions that did not align to the RefSeq viral database correspond to the bacterial host.

unknown-unknown: 4,204,364 contigs with BLASTn hits that did not meet the criteria for 'known-knowns' or 'known-unknown', as well as any contigs where no hits were found. These contigs comprised the vast majority of processed contigs.

Cellular sequences from RefSeq were not utilized to filter out non-viral sequences as most cellular genomes contain one or more proviruses/prophage [6,12]. While salient viral information was obtained from thousands of contigs, the observation that a vast majority of the contigs could not be characterized by this method underscored the need for fast domain-mapping approaches to categorize this type of data. The 'unknown-unknown' contigs from metagenomic samples that did not undergo virus enrichment are likely primarily bacterial and other cellular sequences. However, many novel viral sequences can be expected in this set. Some of the BLAST results were pre-computed and loaded into google's BigQuery, to provide a reference for initial testing during the hackathon. Many of the hackathon participants were not familiar with large scale analysis from databases, and better tutorials on how to leverage cloud infrastructure may be warranted. Therefore, while SQL like tables may be convenient means of presenting data, some additional training is necessary for them to be useful. Finally, while salient viral information was obtained from thousands of contigs, the observation that a vast majority of the contigs could not be characterized by this method underscored the need for fast domain-mapping approaches to categorize this type of data.

Figure 4. DUMMY: Data segmentation figure.

2.4. Data Clustering

The contigs in the set 'unknown-unknown' were clustered to reduce the dataset size and facilitate their further analysis. To this end, all virus RefSeq sequences and the contig sequences were aligned against each other by combined them into one Blast database. This self comparison of XXX contigs yielded XXX query-subject pairs which were treated as edges of a graph with edge weight equal to the log of their E-value. The graph was then clustered via Markov Clustering (MCL, [13]), and the resulting subgraphs analyzed. The distribution of cluster sizes is seen in Figure 5. A total of X clusters were returned, representing an approximately 2-fold reduction in data. Of these clusters, X% were singletons, indicating that the contig was unique among those contigs analyzed. The structure of a few clusters is illustrated in Figure 5. The topology of these graphs is undoubtedly associated with the choice of assembly strategy, as here we took a very conservative approach and expect little to no overlap between contigs from within a single SRR. Thus it is likely that clusters with more complex structure Figure 5a, particularly in cases in which RefSeq sequences aren't acting as a bridge(Figure 5b),

Figure 5. DUMMY: Cluster size histograms and structure. (**a**) Complex cluster structures. (**b**) RefSeq sequences not acting as bridge. (**c**) Cluster hairballs

represent shared genomic content across samples. Similarly, "hairballs", likely represent distinct regions across a single genome, present in only a single SRR (or poorly represented in multiple SRRs, Figure 5c).

Initial investigations explored the use of MMseqs2 [14], Pajek [15], and Gephi [16] to construct and visualize the clusters, but various issues, including issues with reproducing the initial results (discussed more below), precluded presenting those results here. Another challenge with this approach is the computational resources required and the poor scaling with sample size. The blast analysis can be parallelized if the resources are available and one is familiar with how to implement such an approach. Many of the hackathon participants were not familiar with how to do that, and so better tutorials on how to leverage cloud infrastructure may be warranted. Additionally, interpreting such a large volume of BLAST results is non-trivial, and even MCL took 12 hours to cluster the results with 32 cores and 240 GB RAM (interestingly, MCL was found to be unable to effectively take advantage of the full 96 cores made available); thus, if BLAST is to remain a key component of many bioinformatic workflows when working in the cloud with Big Data, additional tools to support the analysis of the results will be beneficial.

2.5. Domain Mapping

Contigs that have been annotated as 'unknown-unknowns' were further classified as described below. In order to get a more nuanced assessment of the genomic content of these contigs, the Conserved Domain Database (CDD) [17] was queried. The entire CDD database was split into 32 parts in order to benefit most from the available threads, and the contigs were analyzed via RPStBLASTn [10] against the fragmented database. Domains with significant hits (e-value < 0.001) were subsequently divided into five bins containing the corresponding Position-Specific Scoring matrices (PSSMs), based on their CDD accession number. These bins were created by using the 'organism' taxonomic information provided with CDD, resulting in a viral bin (2,082 PSSMs), a bacterial bin (19,383 PSSMs), an archaeal bin (1,644 PSSMs), a eukaryotic bin (17,201 PSSMs), and a so-called unknown bin (15,682 PSSMs). To reduce the computational burden downstream, contigs have been filtered based on the taxon-specific PSSMs they carried. Contigs that carried no viral hits and more than three hits to eukaryotic or bacterial CDDs, were excluded from the further analysis.

Out of 347,188 contigs annotated as 'unknown-unknowns' 180820 (52%) were excluded, and 166,368 passed to the downstream analysis (48%). Out of the contigs that passed, 39986 (0.1%) were classified as 'dark matter', i.e. having no hit to any CDD. Most of the excluded contigs had more than 3 bacterial CDD and no viral CDD hits. Overall, subjected contigs had an enrichment for both bacterial and unknown PSSMs in comparison with the other 3 categories (Figure 6). This could be due to the overrepresentation of these PSSMs in the database, but since there is a comparable number of eukaryotic CDDs present, this skewness is more likely a reflection of the input data. Similar to the work on clustering the contigs, RPStBLASTn requires a lot of computational resources and benefits from parallelization. Additionally, the output format of RPStBLASTn also requires some more thought. In the set-up of the analysis, we choose JSON [18] format, since this would be the easiest way to incorporate downstream in the index (vide infra). However,the algorithm itself doesn't allow specification of what exactly is included in this format. Therefore, the output is unnecessarily bulky

Figure 6. DUMMY: Summary domain results

and quickly becomes more than cumbersome to work with. Output flexibility would vastly increase the potential of this output format for this amount of data.

2.6. Gene Annotation

After a general classification based on CDD mapping, between (putative) viral and non-viral, viral contigs were characterized using a modified viral annotation pipeline, VIGA [19]. Briefly, putatively viral contigs have their ORF predicted with Prodigal [20] and annotated against RefSeq Viral Proteins with BLAST [10] and DIAMOND [21]; and search for conserved motifs from pVOGs [17] (prokaryotic virus) and RVDB [22] (all virus-like sequences but not from prokaryotic viruses) using HMMER [23].

Tackling a very large dataset computational efficiency was a concern. While BLAST and DIAMOND can be parallelized to certain degree, HMMER cannot be efficiently parallelized (http://eddylab.org/software/hmmer/Userguide.pdf). To partially mitigate this behaviour, VIGA was parallely invoked from the command line, to run as many instances as CPUs asked, instead of a single instance with all CPUs (https://github.com/NCBI-Hackathons/VirusDiscoveryProject/blob/master/VirusGenes/scripts/director.sh). Each VIGA process was started with only the contigs from a single SRR dataset on a single processor, and later ran 160 such processes in parallel. Initial test runs of 4,400 contigs running on 160 processors showed performance of about 25 sec/contig/processor. In real-time, one million contigs will take approximately 7,000 processor hours. Results from the modified VIGA pipeline provide viral-specific taxonomic/functional annotations to all putative viral contigs, FIGX, based on similarity search by sequence alignment (BLAST and DIAMOND) and modelization (HMMer against pVOG and RVDB). Virus hunting tool kit ("VHT") contig IDs are appended to the VIGA output and putative protein sequences were extracted from the GenBank output. Additionally, viral quotient, the percentage of a pVOG domain created from viral genes, is appended to observations with a hit against the pVOG database.

As noted above, processing such a large volume of data requires massive parallelization, a task which occupied a significant portion of this groups time. Relatedly, interpreting the volume of results provided remains a challenge. Different algorithms may have different computational costs or needs (CPU- vs. memory-expensive process), therefore successful pipelines should fine-tune those needs to the available resources. Combination of different search strategies increases the run time of the pipeline but, if run under an appropriate decision tree, increases the confidence during taxonomical and/or functional annotation.

2.7. Tackling the Unknown

As a large number of contigs remain uncharacterized despite the aforementioned approaches, we aligned 2,527 "unknown unknown" contigs (minimum length size = 1kb) against Virus RefSeq using tBLASTx [10] with default parameters. This post-domain screen revealed hits to phages, Cas-related nucleases and ftsZ-homologs. These results were confirmed in a subsequent analysis using HHPred [24] confirmed these. Identification of these proteins was complicated by the intricate nature of phage genes, their associated bacteria hosts, and the short nature of these contigs (length $<$ 7.6 kb, mean length $=$ 1.5 kb).

The analysis of 4,026 contigs from the 'known-unknown' using VIGA [19] and BLAST [10] revealed one contig of interest which was subsequently identified as a novel norovirus (see Supplementary Material and Methods).

Figure 7. DUMMY: Database design

2.8. Indexing and Metadata

As missing metadata often complicates identifying NGS data sets of interest, we tried to infer metadata information based on SRR contig content. SRRs were clustered using MASH [25], and six main clusters of samples were identified, showing certain diversity in terms of viral content across the dataset. In order to unravel the drivers of that composition-based clustering, the words from the SRA study comments and abstracts were extracted using [26]. A vector of word frequencies was constructed across the selected samples. A PLS was performed in order to identify any co-variance between the identified clusters using MASH and the word frequencies associated to the samples. No strong co-variance could identified using this approach, suggesting that abstracts and comments vocabularies are too vague to automatically caracterize samples.

As a proof of concept, we show that natural language processing (NLP) trained on SRA and associated project metadata can identify SRAs from human gut microbiome metagenomes. Doc2vec [27], an NLP algorithm that uses unsupervised learning to embed variable-length texts into a vector, was trained on the SRA metadata of 628 samples and transformed the metadata into a 300-dimension vector. t-SNE [28], a popular dimensionality reduction tool, was trained and transformed the vectors into coordinates for a 2D space. The SRA metadata was labeled based on the center_project_name, which is typically used to identify the environment from which the metagenome was sequenced from. Three “center_project_name” classes were examined: “human gut microbiome,” “NA”/“None,” and “other.” Figure X shows that all three classes are easily and cleanly separable. Next, NA samples were removed from the dataset and Doc2vec and t-SNE were retrained on this new dataset. In this setting, SRA metadata from human gut microbiome projects can still be distinguished from other projects. Some possible uses of this technique include correcting mislabeled metadata or annotating SRA’s with missing metadata.

To organize the analyzed data we decided to use an indexing scheme implemented in MongoDB. The layout of the data that will be added to the database is expected to be around four tables - SRA metadata, contig description metadata, known contigs information and unknown contig predictions. To add more usability, taxonomy and domain tables will need to be joined hierarchically to the ‘known’ and ‘unknown’ tables (Figure 7). Some of the layout of the data was predicted to do better in a relational database structure as several unrelated data sets must be cross-referenced together in order to support queries.

3. Discussion

Here we present the results from the NCBI's Virus Discovery Hackathon. A diverse group of international researchers met and characterized not only characterized the viral content in 3,000 metagenomic a subset of the SRA datasets, and in doing so,but also identified opportunities to improve apply bioinformatic approaches using cloud computing infrastructure and bioinformatics research to the analysis of analyze NGS datasets. The original intent of the hackathon was to develop an index of SRA run sets that is searchable could be searched based on the viral content contained withinof the runs. To that end, several use cases were identified to guide development.

The use cases developed are outlined below. 1) Identifying shared genomic content across runs. Thus users may submit a sequence, and find all runs from which similar contigs can be derived. 2) Filter based on run metadata. This is essentially the same service provided by the NCBI Entrez Index. 3) Gene/Domain based searches. Users may want to find only runs likely to encode some gene or functional domain of interest, as determined by an analysis of contigs assembled from the runs. 4) Searching based on virus taxonomy. A user may want to find runs likely to contain a particular viral taxa based on an analysis of contigs.

While the data was not quite ready to be indexed, some test data was used to evaluate the database scheme. A full interface for user access will require further development, but testing of particular use cases was made possible through Python notebooks [29] and a collection of API endpoints. Successful query examples were completed for multiple SRA metadata fields, and the information could be obtained in JSON [18] format or returned in tabular format within the notebook approach. To help users who are not fluent in writing database queries or parsing through JSON format, we made use of a PyMongoDB library to run database lookups using python scripts. This requires the user to run the scripts on the same machine where the database is set up, but starting from database lookup to visualization using matplotlib or personal R scripts can all be run on one platform - Jupyter Notebooks [29]. With the complete SRA datasets, the tables will get much larger in terms of the number of entries and the number of fields to describe each dataset. As a result, the relationship between the tables may need to be altered.

Despite generating a number of interesting insights, technical challenges prevented more rapid progress. That said, we feel that these represent opportunities for future development to enhance cloud-based bioinformatic infrastructure and practice. While everyone involved appreciated working with contigs, as opposed to the reads, the sheer volume of SRA data means that the contigs do not represent enough data compression for efficient workflows. While effort was made to identify a test data set, this data set was still perhaps too large, as it represented nearly 55 million contigs. Thus, for future hackathon-type events, especially if the focus is on Big Data, it is recommended that a number of test sets be developed of various sizes, ideally nested such that the smaller sets are subsets of the larger sets, and that they capture the diversity of the full data set as much as is possible. More generally, developing a tool to generate subsamples from arbitrary inputs, relevant to bioinformatic studies, may be useful, not only for testing purposes, but also to allow estimation of how run times scale with sample size for a given computational task or set of tasks. This in turn will support estimating costs.

Jupyter was immensely popular as a framework from which to develop work-flows and conduct exploratory analysis. However, supporting Jupyter in the cloud is not straightforward. Simultaneously supporting collaboration between groups, controlling access to machines, and allowing access to data buckets is challenging. Further efforts are needed to determine which notebooks formats are best suited to the hackathon environment. Relatedly, it was found that, when working at such a large scale, I/O remains a hurdle and workflows developed around BigData analysis in the cloud should accommodate this. Another challenge, felt most acutely by those working on applying machine learning to SRA data is the need for clean metadata. When we spend time curating datasets we should work on the ones with the most metadata, and this should be considered when constructing test data-sets in the future. Additionally, it was found that not all data labeled as WGS appeared to be WGS data, emphasizing the need for better metadata documentation by the research community. The sharing and reuse of data is

one of the primary drivers behind open, FAIR bioinformatic cyberinfrastructure [30]. As discussed above, many SRA entries have incomplete metadata, which deters researchers from performing their own analyses on other scientist's data. Completing the metadata would promote the reusability of data archived in NCBI's databases.

A major goal of this work was to establish domain profiles of NGS data sets, as these have immense potential for supporting sorting and filtering of these massive datasets. They should be treated as first-class reference objects, and a massive expansion of these data objects may be the most effective way to expand into new data spaces. To this end, a follow-up hackathon is currently being planned, during which it is hoped that progress can be made on identifying a Jupyter framework that supports collaborative pipeline development, and which will result in an index of at least a small portion of the metagenomic data set available in the SRA.

4. Materials and Methods

4.1. Participant Recruitment

After initial conception of this project by BB, RJ (JRB?) and RE, RE offered to provide a venue for an international hackathon. Participants were recruited through the outreach efforts of BB, RJ, and RE. VZ identified datasets, which were then parsed by RE using PARTIE [9] to look at any potential amplicon or 16S character. The resulting set of SRRs can be found in Supplemental File 1.

4.2. Assembling contigs from metagenomic datasets (pre-Hackathon)

Contigs containing putative virus sequences were assembled from metagenomic SRA datasets by removing human reads and assembling putative virus sequences into contigs using SKESA [31]. All reads from an SRR archive were aligned against the human genome reference sequence (GRCh38.p12) using HISAT2 [32] (see the associated github repository for execution details, https://github.com/NCBI-Hackathons/VirusDiscoveryProject). Reads mapping fully or partially to the human genome were classified as 'human'. Putative virus sequences in the remaining reads were identified using a K-mer taxonomy approach (NCBI, unpublished). The remaining NCBI taxonomy identifiers were used to extract sequences from Refseq. Given that some viruses are overrepresented in RefSeq, only a few per species were selected at random while for viruses with segmented genomes, e.g. Influenza, all sequences were selected and deduplicated by k-mer distances in a later step using MASH [25]. Putative virus reads were assembled using the guided_assembler from the SKESA and these contigs obtained identifiers based on the guide accessions with a sequential index as a suffix, for example NC_006883.2_1 is based on Prochlorococcus phage genome NC_006883.2.In cases where guide selection failed to detect good reference sets a default viral reference set was used based on the ViralZone database [33].

Reads not classified as virus or human were de-novo assembled with SKESA. For the assembled runs the de novo contigs served as a reference set to align the reads with HISAT2 (as above). The reads that didn't align onto either human, viral or de-novo contigs were classified as unknown. As a result of the workflow each run was re-aligned onto human, viral and de-novo contigs and contains the same set of reads as the original run. The alignments were converted into SRA format without quality scores and stored in google cloud storage for later analysis. Given that most SRA metagenomic reads are bacterial or of unknown origin this step was the most computationally intensive with significant memory and runtime requirements. Due to the limited budget a timeout was introduced on de-novo assembly step and some runs failed to complete.

4.3. Domain mapping

Contigs that were classified by BLASTn as unknown-unknowns, were inspected for domain content using RPSTBPASTN [10]. Briefly, the protein domains from the Conserved Domain Database [17] were downloaded and split into 32 different databases, to benefit most from the available threads. RPStBLASTn searches were ran with an e-value cut-off of 1e-3, and the output was generated in json format. A working example and the commands used are available on GitHub under https://github.com/NCBI-Hackathons/VirusDiscoveryProject/tree/master/DomainLabeling/example

4.4. Megablast

Contigs and Refseq virus nucleotide sequences were stored in a single flat file. Coding-complete, genomic viral sequences were extracted from the NCBI Entrez Nucleotide database (https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?VirusLineage_ss=Viruses,%20taxid:10239&SeqType_s=Nucleotide) to create a specific database using the makeblastdb command-line tool [10]. All sequences were compared against all sequences using MEGABLAST [10] with an E-value cut-off of $1e^{-10}$ and a maximum of one contiguously-aligned region (High-scoring Segment pair, HSP) per query-subject pair.

4.5. Markov Clustering

Markov Clustering (MCL) [13] was applied to blast results as outlined in the associated documentation (https://micans.org/mcl). Briefly, tabular blast output was modified to include only qacc, sacc, and E-value columns, and passed to mcxload to generate network and dictionary files. Thus the set of query and subject pairs is treated as the edge set for a graph, the associated E-values are treated as edge weights. The stream-mirror argument was used to ensure the network is undirected, and stream-neg-log10 and stream-tf 'ceil(200)' arguments were used to log transform E-values, setting a maximum value of 200 for edge weights. Finally, the mcl algorithm was run on the loaded network with an inflation value of 10, and 32 threads. All MCL work was performed on a Google Cloud Platform (GCP) machine with 96 cores and 240 Gb RAM.

4.6. VIGA

Modifications were made to the standard VIGA [19] protocol to enhance the overall speed of the program, removing the rRNA detection step by INFERNAL [34]. This pipeline handled this information, enhancing the identification of viral specific hidden-Markov models (HMM) annotations by the utilization of the complete pVOG database [35] (9,518 HMMs) and the addition of RVDB [22] using HMMER suite [23]. Modified scripts and instructions to reproduce all steps are available on GitHub at https://github.com/NCBI-Hackathons/VirusDiscoveryProject/tree/master/VirusGenes. All viral annotation was performed on a GCP machine with 160 cores and XX Gb RAM

Supplementary Materials: The following are available online at http://www.mdpi.com/2073-4425/xx/1/5/s1, Supplementary Material 1: Dark Matter Methods, Supplementary Material 2: Machine Learning Methods Supplementary Material 3: Index Methods.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing–original draft preparation, X.X.; writing–review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y.", please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflict of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript, or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results".

1. Mardis, E.R. A decade's perspective on DNA sequencing technology. *Nature* **2011**, *470*, 198–203. doi:10.1038/nature09796.
2. Kodama, Y.; Shumway, M.; Leinonen, R.; International Nucleotide Sequence Database Collaboration. The Sequence Read Archive: explosive growth of sequencing data. *Nucleic Acids Res.* **2012**, *40*, D54–D56. doi:10.1093/nar/gkr854.
3. NIH Office of Data Science Strategy. STRIDES. Available online, 2019. Accessed: 2019-07-15.
4. Sayers, E.W.; Agarwala, R.; Bolton, E.E.; Brister, J.R.; Canese, K.; Clark, K.; Connor, R.; Fiorini, N.; Funk, K.; Hefferon, T.; Holmes, J.B.; Kim, S.; Kimchi, A.; Kitts, P.A.; Lathrop, S.; Lu, Z.; Madden, T.L.; Marchler-Bauer, A.; Phan, L.; Schneider, V.A.; Schoch, C.L.; Pruitt, K.D.; Ostell, J. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **2019**, *47*, D23–D28. doi:10.1093/nar/gky1069.
5. Leinonen, R.; Sugawara, H.; Shumway, M.; International Nucleotide Sequence Database Collaboration. The Sequence Read Archive. *Nucleic Acids Res.* **2010**, *39*, D19–D21. doi:10.1093/nar/gkq1019.
6. Roux, S.; Hallam, S.J.; Woyke, T.; Sullivan, M.B. Viral dark matter and virus-host interactions resolved from publicly available microbial genomes. *eLife* **2015**, *4*. doi:10.7554/eLife.08490.
7. Carroll, D.; Daszak, P.; Wolfe, N.D.; Gao, G.F.; Morel, C.M.; Morzaria, S.; Pablos-Méndez, A.; Tomori, O.; Mazet, J.A.K. The Global Virome Project. *Science (New York, N.Y.)* **2018**, *359*, 872–874. doi:10.1126/science.aap7463.
8. Shi, M.; Lin, X.D.; Tian, J.H.; Chen, L.J.; Chen, X.; Li, C.X.; Qin, X.C.; Li, J.; Cao, J.P.; Eden, J.S.; Buchmann, J.; Wang, W.; Xu, J.; Holmes, E.C.; Zhang, Y.Z. Redefining the invertebrate RNA virosphere. *Nature* **2016**, *540*, 539–543. doi:10.1038/nature20167.
9. Torres, P.J.; Edwards, R.A.; McNair, K.A. PARTIE: a partition engine to separate metagenomic and amplicon projects in the Sequence Read Archive. *Bioinformatics (Oxford, England)* **2017**, *33*, 2389–2391. doi:10.1093/bioinformatics/btx184.
10. Camacho, C.; Coulouris, G.; Avagyan, V.; Ma, N.; Papadopoulos, J.; Bealer, K.; Madden, T.L. BLAST+: architecture and applications. *BMC Bioinf.* **2009**, *10*, 421. doi:10.1186/1471-2105-10-421.
11. Brister, J.R.; Ako-Adjei, D.; Bao, Y.; Blinkova, O. NCBI viral genomes resource. *Nucleic Acids Res.* **2015**, *43*, D571–D577. doi:10.1093/nar/gku1207.

12. Liu, H.; Fu, Y.; Xie, J.; Cheng, J.; Ghabrial, S.A.; Li, G.; Peng, Y.; Yi, X.; Jiang, D. Widespread endogenization of densoviruses and parvoviruses in animal and human genomes. *Journal of virology* **2011**, *85*, 9863–9876. doi:10.1128/JVI.00828-11.
13. Enright, A.J.; Van Dongen, S.; Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **2002**, *30*, 1575–1584. doi:10.1093/nar/30.7.1575.
14. Mirdita, M.; Steinegger, M.; Söding, J. MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics (Oxford, England)* **2019**. doi:10.1093/bioinformatics/bty1057.
15. Batagelj, V.; Mrvar, A., Pajek — Analysis and Visualization of Large Networks. In *Graph Drawing Software*; Jünger, M.; Mutzel, P., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2004; pp. 77–103. doi:10.1007/978-3-642-18638-7_4.
16. Bastian, M.; Heymann, S.; Jacomy, M. Gephi: An Open Source Software for Exploring and Manipulating Networks. *AAAI Publications* **2009**.
17. Marchler-Bauer, A.; Bo, Y.; Han, L.; He, J.; Lanczycki, C.J.; Lu, S.; Chitsaz, F.; Derbyshire, M.K.; Geer, R.C.; Gonzales, N.R.; Gwadz, M.; Hurwitz, D.I.; Lu, F.; Marchler, G.H.; Song, J.S.; Thanki, N.; Wang, Z.; Yamashita, R.A.; Zhang, D.; Zheng, C.; Geer, L.Y.; Bryant, S.H. CDD/SPARCLE: functional classification of proteins via subfamily domain architectures. *Nucleic Acids Res.* **2017**, *45*, D200–D203. doi:10.1093/nar/gkw1129.
18. Bray, T. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 7159, RFC Editor, 2014.
19. González-Tortuero, E.; Sutton, T.D.; Velayudhan, V.; Shkoporov, A.N.; Draper, L.A.; Stockdale, S.R.; Ross, R.P.; Hill, C. VIGA: a sensitive, precise and automatic de novo VIral Genome Annotator. *bioRxiv* **2018**, p. 277509. doi:10.1101/277509.
20. Hyatt, D.; Chen, G.L.; Locascio, P.F.; Land, M.L.; Larimer, F.W.; Hauser, L.J. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinf.* **2010**, *11*, 119. doi:10.1186/1471-2105-11-119.
21. Buchfink, B.; Xie, C.; Huson, D.H. Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* **2015**, *12*, 59–60. doi:10.1038/nmeth.3176.
22. Goodacre, N.; Aljanahi, A.; Nandakumar, S.; Mikailov, M.; Khan, A.S. A Reference Viral Database (RVDB) To Enhance Bioinformatics Analysis of High-Throughput Sequencing for Novel Virus Detection. *mSphere* **2018**, *3*. doi:10.1128/mspheredirect.00069-18.
23. Eddy, S.R. HMMER: biosequence analysis using profile hidden Markov models. Accessed: 2019-07-15.
24. Hildebrand, A.; Remmert, M.; Biegert, A.; Söding, J. Fast and accurate automatic structure prediction with HHpred. *Proteins* **2009**, *77 Suppl 9*, 128–132. doi:10.1002/prot.22499.
25. Ondov, B.D.; Starrett, G.J.; Sappington, A.; Kostic, A.; Koren, S.; Buck, C.B.; Phillippy, A.M. Mash Screen: High-throughput sequence containment estimation for genome discovery. *BioRxiv* **2019**, p. 557314. doi:10.1101/557314.
26. Zhu, Y.; Stephens, R.M.; Meltzer, P.S.; Davis, S.R. SRAdb: query and use public next-generation sequencing data from within R. *BMC Bioinf.* **2013**, *14*, 19. doi:10.1186/1471-2105-14-19.
27. Le, Q.V.; Mikolov, T. Distributed Representations of Sentences and Documents. *arXiv e-prints* **2014**, [http://arxiv.org/abs/1405.4053v2].
28. van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **2008**, *9*, 2579–2605.
29. Jupyter Steering Council. The Jupyter/IPython Project. Online. Accessed: 2019-07-15.
30. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; Bouwman, J.; Brookes, A.J.; Clark, T.; Crosas, M.; Dillo, I.; Dumon, O.; Edmunds, S.; Evelo, C.T.; Finkers, R.; Gonzalez-Beltran, A.; Gray, A.J.G.; Groth, P.; Goble, C.; Grethe, J.S.; Heringa, J.; 't Hoen, P.A.C.; Hooft, R.; Kuhn, T.; Kok, R.; Kok, J.; Lusher, S.J.; Martone, M.E.; Mons, A.; Packer, A.L.; Persson, B.; Rocca-Serra, P.; Roos, M.; van Schaik, R.; Sansone, S.A.; Schultes, E.; Sengstag, T.; Slater, T.; Strawn, G.; Swertz, M.A.; Thompson, M.; van der Lei, J.; van Mulligen, E.; Velterop, J.; Waagmeester, A.; Wittenburg, P.; Wolstencroft, K.; Zhao, J.; Mons, B. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* **2016**, *3*, 160018. doi:10.1038/sdata.2016.18.
31. Souvorov, A.; Agarwala, R.; Lipman, D.J. SKESA: strategic k-mer extension for scrupulous assemblies. *Genome Biol.* **2018**, *19*, 153. doi:10.1186/s13059-018-1540-z.

534 32. Kim, D.; Langmead, B.; Salzberg, S.L. HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods* **2015**, *12*, 357–360. doi:10.1038/nmeth.3317.
536 33. Hulo, C.; de Castro, E.; Masson, P.; Bougueleret, L.; Bairoch, A.; Xenarios, I.; Le Mercier, P. ViralZone: a knowledge resource to understand virus diversity. *Nucleic Acids Res.* **2011**, *39*, D576–D582. doi:10.1093/nar/gkq901.
539 34. Nawrocki, E.P.; Eddy, S.R. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics (Oxford, England)* **2013**, *29*, 2933–2935. doi:10.1093/bioinformatics/btt509.
541 35. Grazziotin, A.L.; Koonin, E.V.; Kristensen, D.M. Prokaryotic Virus Orthologous Groups (pVOGs): a resource for comparative genomics and protein family annotation. *Nucleic Acids Res.* **2017**, *45*, D491–D498. doi:10.1093/nar/gkw975.
544 36. Shean, R.C.; Makhsous, N.; Stoddard, G.D.; Lin, M.J.; Greninger, A.L. VAPiD: a lightweight cross-platform viral annotation pipeline and identification tool to facilitate virus genome submissions to NCBI GenBank. *BMC Bioinf.* **2019**, *20*, 48. doi:10.1186/s12859-019-2606-y.
547 37. Choi, I.; Ponsero, A.J.; Bomhoff, M.; Youens-Clark, K.; Hartman, J.H.; Hurwitz, B.L. Libra: scalable k-mer–based tool for massive all-vs-all metagenome comparisons. *GigaScience* **2018**, *8*. doi:10.1093/gigascience/giy165.

Supplemental Materials: NCBI's Virus Discovery Hackathon: Engaging Research Communities to Identify Cloud Infrastructure Requirements

Supplementary Material 1. Dark Matter Methods

Contigs designated as 'known_unknonw' were analyzed for potential novel viruses. The contigs were screened for open reading frames (ORFs) and hidden Markov Models (HMMs) using VIGA [S19]. VIGA was deployed with the following virus databases: pVOG [S35], RVDB [S22], and Virus RefSeq [S11]. The tabular VIGA output was converted into a local SQLite database to facilitate its exploration using SQL queries and to export the results in JSON [S18] for indexing (DarkMatter1/tools/make-vigodb/src/mk-vigodb.py, FIG Z).

The VIGA output was extended by calculating the virus quotient for each VIGA hit. The virus quotient is defined by the percentage of viral genes used to train the HMM amongst total genes. To better assess the predicted VIGA annotations, we developed a simple scoring system for the BLASTx and DIAMOND results from VIGA: $\frac{similarity_{reported}+coverage_{reported}}{200} - evalue_{reported}$. A score of 1 indicates the predictions is identical to the templates used by VIGA while a score of 0 indicates no templates were found by VIGA. The VIGA SQLite database was queried for "eukaryotic virus" in the annotations column and contigs with high scores and "virus" in the description line were aligned against the non-redundant database using BLASTn and BLASTx [S10] to further identify these contigs. Contigs were annotated into a GenBank-ready format using VAPiD [S36].

```
"Contig_213_101.679:1.6825_3" : {                  // Entry ID or primary key
  "Contig" : "Contig_213_101.679:1.6825",          // Prediction template
  "Protein_id" : "Contig_213_101.679:1.6825_3",    // Protein | ORF ID
  "Start" : 1576,                                  // Predicted start position [nt]
  "Stop" : 2980,                                   // Predicted stop position [nt]
  "Strand" : 1,                                    // Direction: 1=fwd, -1=rev
  "Length_aa" : 467,                               // Length of predicted protein [aa]
  "Orflength" : 1401,                              // Length of predicted protein [nt]
  "pI" : 6.09124755859,                            // Estimated Isoelectric point
  "Molecular_weight" : 53.5625,                    // Estimated molecular weight [kDa]
  "Instability_index" : 36.6983083512,             // Estimated Instability Index
  "SRR" : "SRR6659510",                            // SRR accession of contig
  "Organism" : "Edwardsiella phage PEi21",         // Source organism of prediction source
  "Hitscore" : 0.736478082660833,                  // Score of prediction
  "Virus_quotient" : 0.832707055,                  // Viral quotient
  "Description" : "phage terminase, large subunit",
  "Sources" : {                                    // Sources used in VIGA
    "VOG4544" : {                                  // HMM hit (pVOG or RVDBs)
        "Evalue" : 7.4e-28,                        // Evalue/Significance of hit
        "Coverage" : 70.1754385965,                // Coverage to closest reference [%]
        "Similarity" : 56                          // Similarity to closest reference [%]
    },
    "DIAMOND" : {                                  // Prediction source
        "Coverage" : 98.07,                        // Coverage to closest reference [%]
        "Similarity" : 51.1,                       // Similarity to closest reference [%]
        "Evalue" : 1.5e-119                        // Evalue/Significance of hit
    }
  }
}
```

Figure S1. Example of annotated domain output in JSON format.

Supplementary Material 2. Machine Learning Methods

Jaccard distance was estimated on the identified viral contigs using MASH [S25]. MASH provides the user with a rapid method to reduce DNA sequences in representative k-mer sketches that are used to estimate a Jaccard similarity between samples. The tool was shown to be scalable in both number of considered metagenomes and size of the samples. MASH was shown to be a reliable tool to cluster amplicon datasets, metagenome read datasets and contig datasets [S37].

A kmer size of 21bp was chosen with a sketch size of 10,000. Samples containing less than two viral contigs were removed from the analysis. A total of 511 samples were kept for the analyzed and clustered by ward clustering. A manual cleaning of the terms was performed to remove punctuation and low-informative terms. In total, 210 samples with abstract and comments were analyzed.

Supplementary Material 3. Index Methods

Setting up test data and databases: Three virtual machines were setup with Solr, MongoDb and PostgreSQL. Test tables were created through running a bigquery generate test tables with n

entries to test the three databases. To generate these tables, we used a script saved in GitHub at: https://github.com/NCBI-Hackathons/VirusDiscoveryProject/blob/master/ScalableIndex. Three test tables were generated with 100 entries, and one million entries for SRA metadata, contig description and known contigs metadata. This was done without optimization or indexing.

Uploading test data to the database: To upload the data to Solr, we used a schemaless document upload with the Solr interface. A script was used to import the JSON files to MongoDB (uploaded to GitHub), and another script was written to upload the data to PostgreSQL. PostgreSQL was the only database where the data types for each field needs to be defined. Overall, the data import for all three databases took less than a minute for the different databases and the test datasets generated.

Indexing the field: Solr indexes all the fields in the document when uploaded as a schemless import. No fields were indexed for MongoDB and PostgreSQL with the current setup and files. This could have serious performance consequences, as the lookup complexity for non-indexed sql is usually considered on the order of n, where n is the number of elements in the database. For searching where an indexed field is used as the primary clause, the complexity is $n * log(n)$.