Spfy: an integrated graph-based sequence database for real-time prediction of Escherichia coli phenotypes

Kevin K Le*[1], Matthew D Whiteside[1], James Hopkins[1], Victor PJ Gannon[1] and Chad R Laing†[1]

[1]National Microbiology Laboratory at Lethbridge, Public Health Agency of Canada, Twp Rd 9-1, Lethbridge, AB, T1J 3Z4, Canada

February 21, 2018

Abstract

Current comparative computational workflows chain different analysis software, but lack storage and retrieval methods for generated results. Spfy uses a graph database to store and retrieve results for computational workflows, facilitating the management and querying of tens of thousands of whole-genome *Escherichia coli* sequences, and efficient downstream processing. By making the storage and retrieval of results part of the platform, with data effectively linked to the organisms of interest through a standardized ontology, we can facilitate inquiries ranging from population genomics to epidemiological investigations, and mitigate the recomputing of analyses. A graph approach is flexible to accommodate new analysis software as they are developed. Integrated data storage will be necessary as publicly available whole genome sequencing data for bacterial pathogens currently numbers in the tens of thousands, with hundreds of thousands set to be available within the next few years.

Database URL: `https://lfz.corefacility.ca/superphy/spfy/`.

1 Introduction

Whole genome sequencing (WGS) can in theory provide the entire genetic content of an organism. This unparalleled resolution and sensitivity has recently transformed public-health surveillance and outbreak response [1, 2]. Additionally, the identification of novel disease mechanisms [3, 4], and rapid clinical diagnoses and reference lab tests based on the specific mechanism of disease are now possible. [5, 6].

The rapid characterization and comparison of bacterial pathogens relies principally on the combination of outputs from multiple software programs that are targeted for specific applications. Examples include the Resistance Gene Identifier (RGI) [7] for antimicrobial resistance (AMR) gene prediction, VirulenceFinder [8] for virulence factor (VF) annotation, and Prokka for bacterial genome annotation with external tools [9]. In particular, RGI and VirulenceFinder represent a series of *in-silico* methods which have been developed to replicate the results of traditional wet-lab tests; this allows new WGS results to be viewed in the context of historical tests.

Comprehensive platforms that combine individual programs into a cohesive whole also exist. These include free platforms such as the Bacterium Analysis Pipeline (BAP) [10], the Integrated Rapid Infectious Disease Analysis (IRIDA) project `http://www.irida.ca/`, and the Pathosystems Resource Integration Center (PATRIC) [11]. Commercial applications, such as Bionumerics, which is used by PulseNet International for the analyses of WGS data in outbreak situations also exist, and offer support as well as accredited, standardized tests [12].

*kevin.le@canada.ca

†chad.laing@canada.ca

Traditionally, these platforms have been applied to samples numbering only a few dozen while WGS data for bacterial pathogens of public health importance have recently accumulated in public databases in the tens of thousands, with hundreds of thousands set to be available within the next few years. For *Escherichia coli*, there are over sixty thousand publicly available genomes in EnteroBase `https://enterobase.warwick.ac.uk/` and three million whole genomes in GenBank [13].

Many of the comparative analyses that are run are broadly useful, and therefore computed multiple times. To begin to compute results on all available genomes, an effective method to mitigate the recomputing of analyses is to make the storage and retrieval of results part of the platform, and effectively linked to the organisms of interest with a standardized ontology. Such measures can help ensure the rapid response times required for public health applications, and allow results to be integrated and progressively updated as new data becomes available.

We have previously developed Superphy [14], an online predictive genomics platform targeting *E. coli*. Superphy integrates pre-computed results with domain-specific knowledge to provide real-time exploration of publicly available genomes. While this tool has been useful for the thousands of pre-computed genomes in its database, the current pace of genome sequencing requires real-time predictive genomic analyses of tens-, and soon hundreds-of-thousands of genomes, and the long term storage and referencing of these results, something that the original SuperPhy platform was incapable of.

In this study, we present the Spfy platform, which integrates a graph database with real-time analysis options. By integrating analysis options with data storage, we avoid recomputing analysis for identical runs. Graph-based result storage also allows retrospective comparisons as more genomes are sequenced or populations change, and is flexible to accommodate new analysis modules as they are developed. The database is available at `https://lfz.corefacility.ca/superphy/spfy/`.

2 FUNCTIONALITY

By supporting multiple *in-silico* subtyping options, the platform functions similar to a reference laboratory, while adding support for population analyses. Subtyping options for *E.coli* are O-antigen, H-antigen, and VF gene determination using ECTyper `https://github.com/phac-nml/ecoli_serotyping`, Shiga-toxin 1 (Stx1), Shiga-toxin 2 (Stx2), and Intimin typing using Phylotyper [15], and AMR annotation using the RGI program [7]. Spfy also performs bioinformatics analyses: pangenome generation using Panseq [16] and support vector machine (SVM)-backed AMR predictions for OmniLog AMR assays using Scikit-learn [17].

These tasks are divided into subtasks, and distributed across a built-in task queue. Results are converted to individual graphs and stored within a larger graph database. By integrating task distribution with graph storage, Spfy enables large-scale analyses, such as epidemiological associations between specific genotypes, biomarkers, host, source, and other metadata, and statistical significance testing of genome markers for user-defined groups. Any data type or relation in the graph is valid for analysis, such as the presence or absence of pan-genome regions against determined serotypes or other subtyping options. In addition, Spfy supports the submission of user-specified metadata, such as location or source information.

3 IMPLEMENTATION

Spfy's server-side code, graph generation, and analysis modules are developed in Python and the front-end website is developed using the React JavaScript library `https://facebook.github.io/react/`. For the addition of new data to the database, the following steps are taken:

i) The upload begins through the website, where user-defined analyses options are selected. The results of these chosen analyses are immediately reported to the user following their completion, while the remaining analyses are subsequently completed and stored in the database without interaction from the user. The public web service accepts up to 200 MB of genome files (50 genomes uncompressed, or 120 genomes compressed) at a time, though an unlimited amount of data can be submitted locally.

ii) User-selected analyses are enqueued into the Redis Queue `http://python-rq.org/` task queue. Redis Queue consists of a Redis Database `https://redis.io/` and task queue workers which run as Python processes.

iii) The workers dequeue the analyses, run them in parallel, and temporarily store results in the Redis database.

iv) Python functions parse the results and permanently store them in Blazegraph `https://www.blazegraph.com/`, the graph database used for Superphy.

3.1 Data Storage

Semantic web technologies describe the relations between different data [18]. For biological data, individual data points such as genome, contiguous DNA sequence, or gene, can be linked together in a searchable graph structure. This allows novel data to be incorporated into the existing graph, and the use of graph databases for semantic information has been proposed as a open standard for sharing public information [19]. Data points are annotated with terms from published ontologies. To avoid proliferating ontologies, and to allow Spfy to integrate with existing ones, we use annotations from the GenEpiO [20], FALDO [21], and TypOn [22].

The permanent storage of results is as a one-time cost to avoid recomputation when the same analysis is re-run. For population analyses, Spfy searches the graph for all data points annotated with the queried ontology term. This graph data is then converted into the required structure, usually numerical arrays, as required for the given analysis modules. In a graph database, search begins at any node or attribute. This is in contrast to a SQL database which requires a predefined schema, or a NoSQL database which treats data as documents with varying structure. For example, the addition of a new analysis module would typically require a new table definition in a SQL database, or the addition of a new document type in a NoSQL database. A graph database instead adds new nodes or attributes which can be connected to existing data, so explicit joins or data conversions are not required. Spfy also adds the graph results in parts, and allows the database to infer their connections.

3.2 Web design

The front-end website is written as a single-page application. To ensure a familiar user interface, we followed the Material Design specification `https://material.io/`, published by Google, surrounding a card-based design. (see Figure 2) Both the task selection and result displays follow the same design pattern: while data storage is graph-based, the results of various analysis modules are presented to users in a familiar tabular structure and available for download as .csv spreadsheet files. (see Figure 3)

No account creation is required to use the platform. A sharable token is automatically created for users upon entering the website, and is embedded into the website address. Users can share results by copying their URL, and files submitted from different computers using the same token will be visible to anyone one the same link.

3.3 Service Virtualization

Docker `https://www.docker.com/` is a virtualization technology to simulate self-contained operating systems on the same host computer, without the overhead of full hardware virtualization [23]. The Spfy platform depends on a series of webservers, databases, and task workers and uses Docker to compartmentalize these services which are then networked together using Docker-Compose `https://docs.docker.com/compose/`. (see Figure 4) Docker integration ensures that software dependencies, which are typically manually installed [9, 16, 24, 25], are instead handled automatically.

One of the key benefits of using more common-place technologies is the compatibility with other infrastructure resources. Docker containers are widely supported by cloud computing services: Amazon Web Services (AWS) `https://aws.amazon.com/docker/`, Google Cloud Platform (GCloud) `https://cloud.google.com/container-engine/`, and Microsoft Azure `https://azure.microsoft.com/en-us/services/container-service/`, and self-hosted cloud computing technologies such as OpenStack `https://wiki.openstack.org/wiki/Docker` all support Docker. Spfy packages compute nodes as reproducible Docker

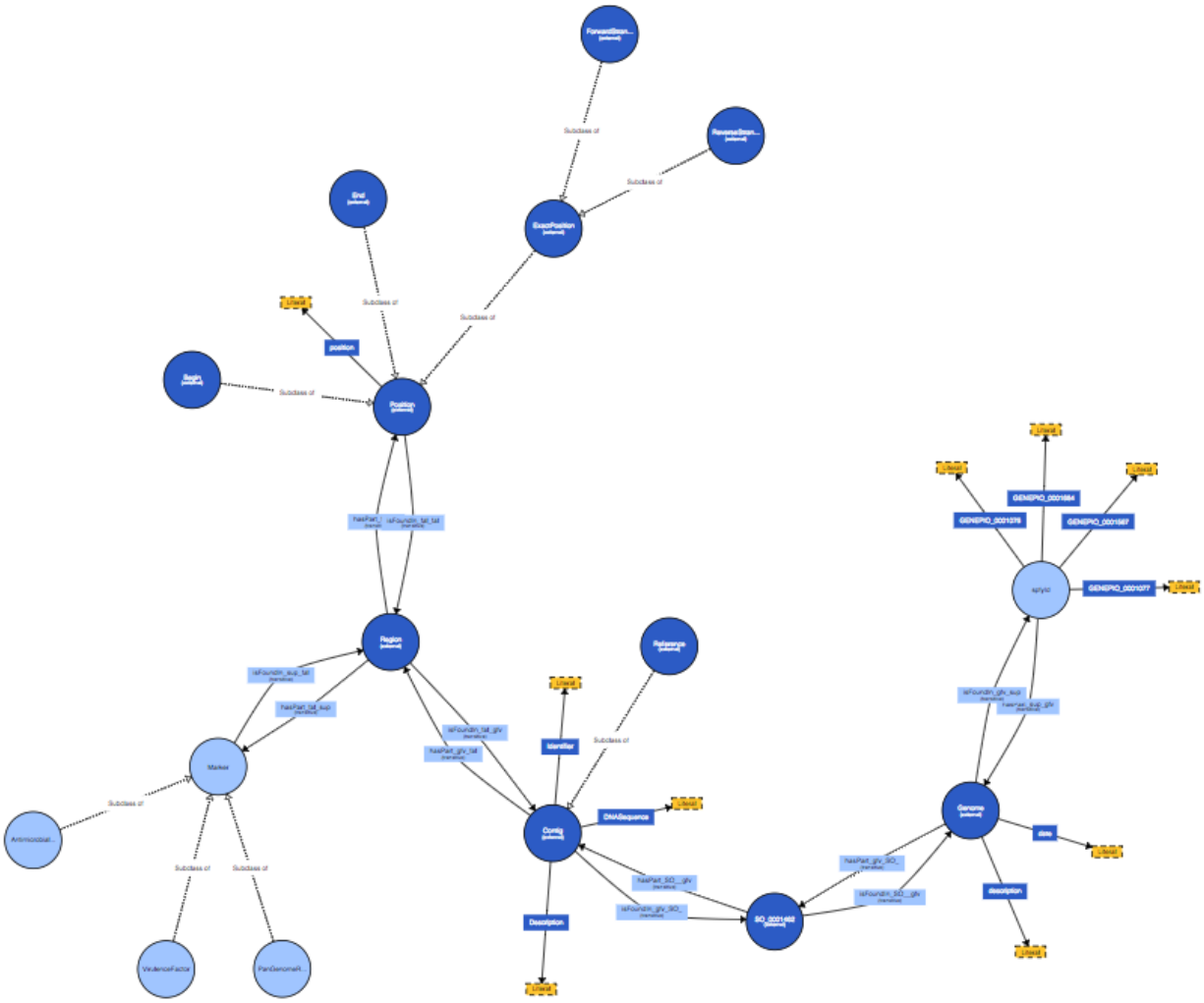


Figure 1: An example of how data is stored in Spfy. Brackets highlight the source of different data points and the software it was generated from. These parts are added in as the analyis modules complete, at varying times, and the overall connections are inferred by the database.

containers, and allows the platform to easily scale to demand.

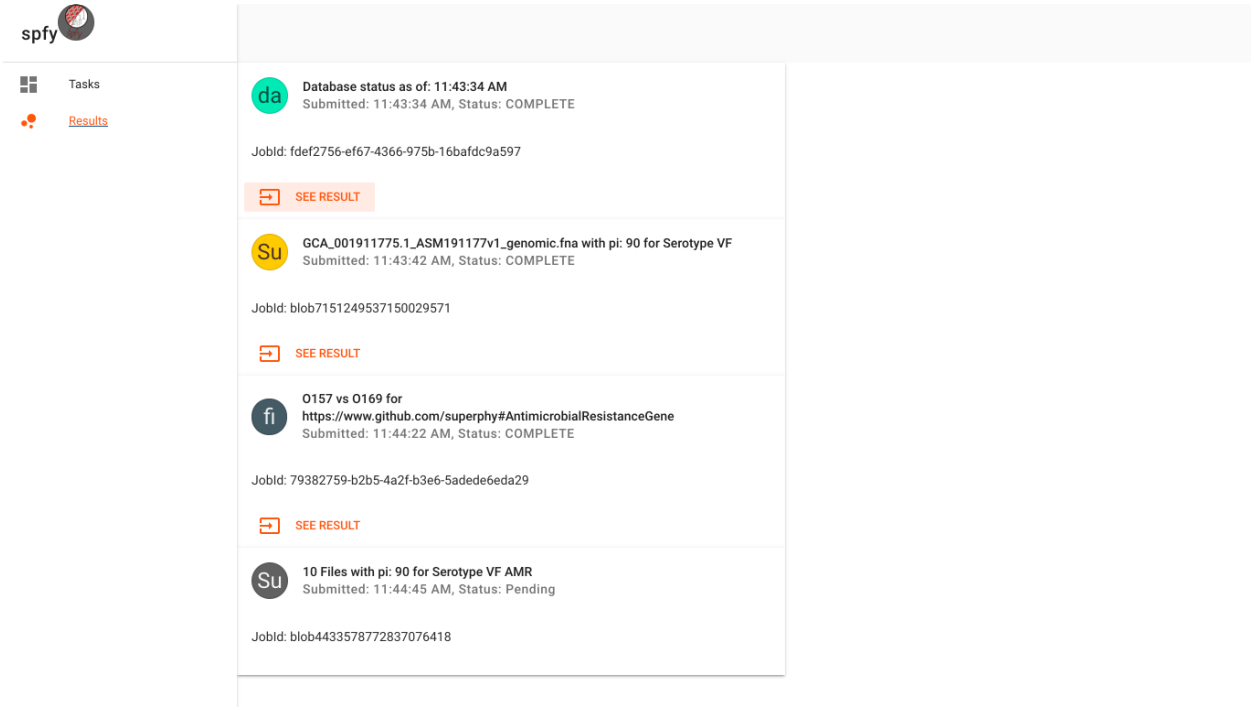
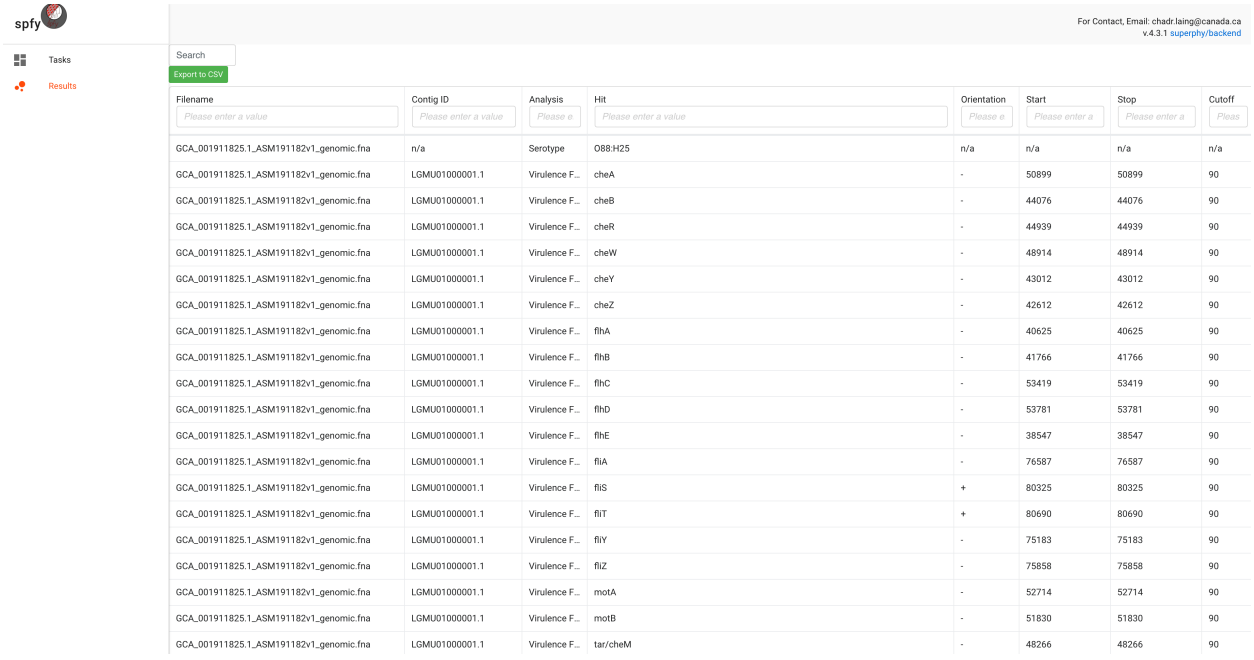


Figure 2: Caption for figure within column.



Filename	Contig ID	Analysis	Hit	Orientation	Start	Stop	Cutoff
GCA_001911825.1_ASM191182v1_genomic.fna	n/a	Serotype	O88:H25	n/a	n/a	n/a	n/a
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	cheA	-	50899	50899	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	cheB	-	44076	44076	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	cheR	-	44939	44939	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	cheW	-	48914	48914	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	cheY	-	43012	43012	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	cheZ	-	42612	42612	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	flhA	-	40625	40625	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	flhB	-	41766	41766	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	flhC	-	53419	53419	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	flhD	-	53781	53781	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	flhE	-	38547	38547	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	fliA	-	76587	76587	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	fliS	+	80325	80325	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	fliT	+	80690	80690	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	fliY	-	75183	75183	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	fliZ	-	75858	75858	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	motA	-	52714	52714	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	motB	-	51830	51830	90
GCA_001911825.1_ASM191182v1_genomic.fna	LGMU01000001.1	Virulence F...	tar/cheM	-	48266	48266	90

Figure 3: Caption for figure within column.

3.4 Continuous integration

Our tests for functionality and backwards compatibility run on TravisCI `https://travisci.io`, a continuous integration (CI) platform. The individual tests use PyTest `https://doc.pytest.org/`, and the current build status can be checked either on our GitHub repository or at `https://travis-ci.org/superphy/backend`. TravisCI also builds Spfy's core Docker images, and uploads them to Docker Hub `https://hub.docker.com/u/superphy/`.

4 RESULTS

Spfy was tested with 10,243 public *E. coli* assembled genomes from Enterobase, storing every sequence and the results for all included analysis modules. All subtyping options: O-antigen, H-antigen, Shiga-toxin 1, Shiga-toxin 2, and Intimin typing, VF and AMR annotation were ran on the test set. The genomes were also analyzed within the pan-genome framework of *E. coli*. The resulting database had 17,820 nodes and 3,811,473 leaves, with 1,125,909,074 object properties. Spfy has been up since May 2017. The server accepts assembled *E. coli* genomes with the *.fasta* or *.fna* extensions. Submissions are checked against a reference set of *E. coli* gene sequences before running analyses.

5 DISCUSSION

Many bioinformatics software programs have been developed *ad hoc*, with individual researchers and laboratories developing software specific to their environment [26]. Such tools were often script-based, with custom data formats, and only suitable for small collections of data [26]. Recent efforts [27, 10] have focused on providing a common web

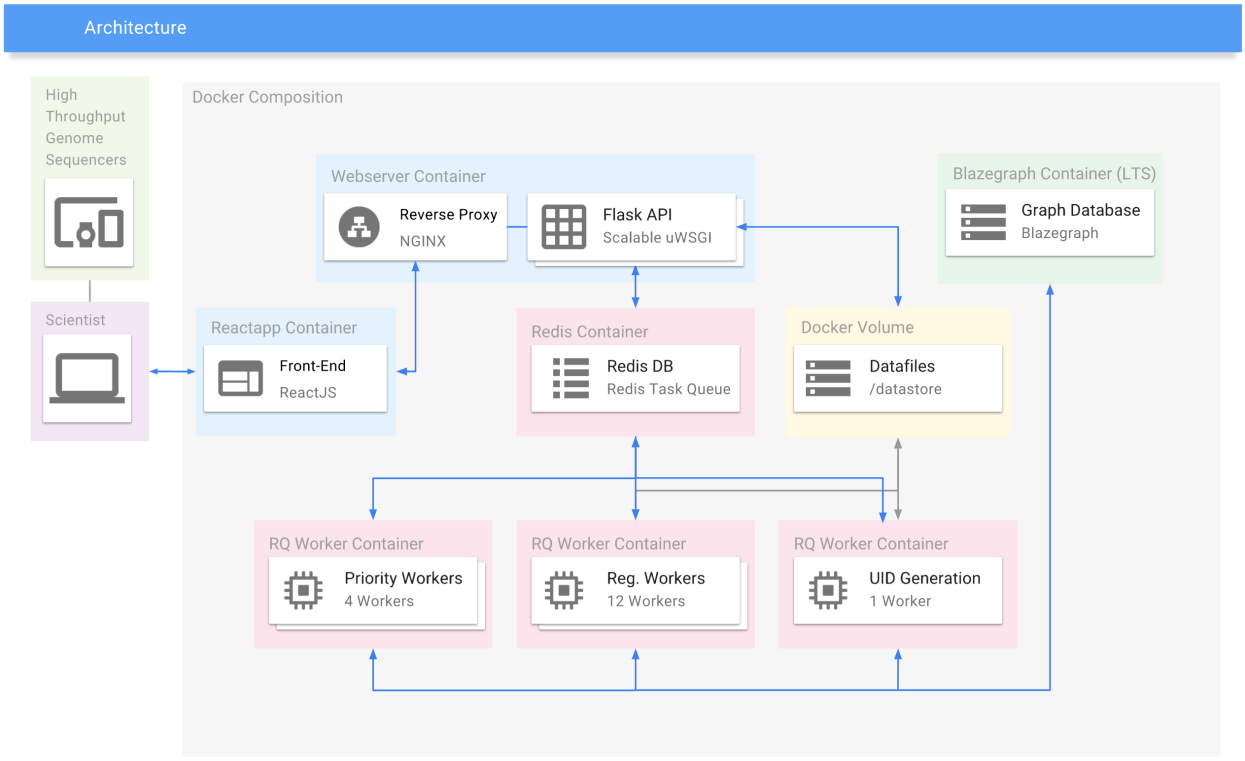


Figure 4: Caption for figure within column.

interface for these programs, while still returning the same result files. However, many subsets of biology now require the analyses of big-data, where inputs are taken from a variety of analysis programs, and involve large-scale data warehousing [28]. The ability to perform computations in real-time, store data in flexible databases, and utilize a common application programming interface (API) linking resources are required for these analyses [29].

One of the key goals in developing Spfy was to accommodate a variety of result formats, from storing subtyping data to retrieving results as inputs for downstream analyses, such as statistical significance testing. We've shown how a graph database can accommodate a variety of programs used by bioinformaticians and is performant for data retrieval on the results of 10,000+ genomes. Spfy maintains instantaneity, as modern analytics platforms must respond to user-requested analyses of big-data in the same efficiency as old analyses of single files.

5.1 Impact on Public Health Efforts

The isolation and characterization of bacterial pathogens are critical for Public Health laboratories to rapidly respond to outbreaks, and to effectively monitor known and emerging pathogens through surveillance programs. Until recently, public-health agencies relied on laboratory tests such as serotyping, pulsed-field gel electrophoresis (PFGE) banding patterns, or Polymerase Chain Reaction (PCR)-based amplification to identify known VFs or AMR genes, along with other tests [1], to characterize bacterial isolates in outbreak and surveillance settings.

AMR testing, VF testing, and Shiga-toxin testing are of particular importance to surveillance efforts due to their role in a pathogen's lethality. Current efforts are focused on predictive genomics, where the relevant phenotypic information can be determined through examination of the whole-genome sequence. WGS methods allow a single laboratory method - the sequencing step - to provide a variety of analysis options, such as AMR, VF, and Shiga-toxin results, and as such can be used to evaluate the spread of outbreaks with better resolution and context than traditional methods [1].

Spfy uses WGS results. After initial sequencing of new isolates, Spfy can be used in place of a traditional reference laboratory, to determine the O-type and H-type, Stx type, and all known VFs and AMR genes in real-time. These results can be shared with other agencies and researchers over the internet. Furthermore, using Spfy's database of pre-processed genomes, Spfy can determine all strains a sample may be related to which is useful for monitoring the evolution of pathogens over time. A computational approach saves the time and cost associated with performing multiple tests per sample, and allows population comparisons to keep pace the current generation WGS data.

5.2 Comparison with other bioinformatic pipeline technologies

Existing scientific workflow technologies such as Galaxy [27], and pipelines such as the Integrated Rapid Infectious Disease Analysis (IRIDA) platform `http://www.irida.ca/` and the Bacterium Analysis Pipeline (BAP) [10] help automate the use of WGS data. Reproducibility is important, and Galaxy aims to provide a reproducible, computation-based research environment which is accessible to individuals without programming knowledge. Galaxy defines a formal schema for linking different analysis software together, so the entire analysis pipeline can be replicated. Building on top of the Galaxy framework, IRIDA adds prebuilt pipelines specific to bioinformatics uses, as well as sequence and result storage. IRIDA takes a project-based approach, with sequences stored per project, and results stored linearly per whole-genome sequence. Similarly, BAP provides an integrated analysis pipeline for bacterial WGS data as a web service. For result storage, these workflow technologies either use relational tables [27], or store resulting files to disk [10].

Table 1: My caption

name	indexType	m	height	nnodes	nleaves	nentries	nodeBytes	leafBytes	totalBytes
__globalRowStore	BTree	32	1	1	6	102	193	8537	8730
kb.lex.BLOBS	BTree	692	2	17	6727	3227686	90596	60187016	89331537662
kb.lex.ID2TERM	BTree	905	2	88	39912	18080577	515355	293436043	2058798455
kb.lex.TERM2ID	BTree	193	3	1153	147978	18080577	5596557	1152764154	1158360711
kb.spo.JUST	BTree	284	3	13213	2042532	299426518	77979362	15527483178	15605462540
kb.spo.OSP	BTree	708	3	1649	639325	262364538	11448125	3760927987	3772376112
kb.spo.POS	BTree	990	2	864	463188	262364538	8347594	2246478879	2254826473
kb.spo.SPO	BTree	1024	2	835	471805	262364538	10308603	2661857997	2672166600

Like IRIDA and BAP, Spfy automates workflows for users, and like Galaxy, Spfy uses task queues to distribute selected analysis. On a per file basis, Spfy performs at a similar speed to BAP on predictive genomics tasks, though Spfy does not provide genome assembly services. Spfy processes XXXX files over XX tasks in XXXX time, distributing computations over a task queue and multiple Docker compartmentalized containers. We also focus on the data warehousing problem in bioinformatics: storing results as in IRIDA and BAP, but using a graph database to integrate results for further analyses. Like BAP, Spfy integrates different programs, such as for VF and AMR determination, into one pipeline.

Because output from these programs is user-specific or transitory, results from identical comparisons are often re-computed. Additionally, output from different analyses are structured using distinct terminology and formats, which must be converted before they can be compared. Without a unified structure, these conversions quickly become impractical for broad usage. Graph-based storage of all results solves these problems and allows Spfy to perform population-wide analyses, regardless of the individual analysis software used to generate the datasets.

6 CONCLUSIONS

Future work will focus on adding additional analyses modules to aid genotype to phenotype predictions, and supporting different species. We're currently working on an approach using support vector machines to predict *E.coli* phenotypes `https://github.com/superphy/kmer` which will be integrated into Spfy. While the integrated approach of storing and retrieving results provides enormous benefits, the developed analyses modules are self-contained and be transfered to existing platforms such as Galaxy, and IRIDA. Spfy's code is hosted at `https://github.com/superphy/backend`, and is available for free under the open-source Apache 2.0 license. A developer guide is provided at `https://superphy.readthedocs.io/en/latest/`.

Conflict of interest. None declared.

7 ACKNOWLEDGEMENTS

References

[1] J Ronholm, Neda Nasheri, Nicholas Petronella, and Franco Pagotto. Navigating microbiological food safety in the era of whole-genome sequencing. *Clinical Microbiology Reviews*, 29(4):837–857, 2016.

[2] Birgitta Lytsy, Lars Engstrand, Åke Gustafsson, and Rene Kaden. Time to review the gold standard for genotyping vancomycin-resistant enterococci in epidemiology: Comparing whole-genome sequencing with pfge and mlst in three suspected outbreaks in sweden during 2013–2015. *Infection, Genetics and Evolution*, 2017.

[3] Kai Wang, Siu Tsan Yuen, Jiangchun Xu, Siu Po Lee, Helen HN Yan, Stephanie T Shi, Hoi Cheong Siu, Shibing Deng, Kent Man Chu, Simon Law, et al. Whole-genome sequencing and comprehensive molecular profiling identify new driver mutations in gastric cancer. *Nature genetics*, 46(6):573–582, 2014.

[4] Ryan KC Yuen, Bhooma Thiruvahindrapuram, Daniele Merico, Susan Walker, Kristiina Tammimies, Ny Hoang, Christina Chrysler, Thomas Nalpathamkalam, Giovanna Pellecchia, Yi Liu, et al. Whole-genome sequencing of quartet families with autism spectrum disorder. *Nature medicine*, 21(2):185–191, 2015.

[5] Laurel K Willig, Josh E Petrikin, Laurie D Smith, Carol J Saunders, Isabelle Thiffault, Neil A Miller, Sarah E Soden, Julie A Cakici, Suzanne M Herd, Greyson Twist, et al. Whole-genome sequencing for identification of mendelian disorders in critically ill infants: a retrospective analysis of diagnostic and clinical findings. *The Lancet Respiratory Medicine*, 3(5):377–387, 2015.

[6] Frederick E Dewey, Megan E Grove, Cuiping Pan, Benjamin A Goldstein, Jonathan A Bernstein, Hassan Chaib, Jason D Merker, Rachel L Goldfeder, Gregory M Enns, Sean P David, et al. Clinical interpretation and implications of whole-genome sequencing. *Jama*, 311(10):1035–1045, 2014.

[7] Andrew G McArthur, Nicholas Waglechner, Fazmin Nizam, Austin Yan, Marisa A Azad, Alison J Baylay, Kirandeep Bhullar, Marc J Canova, Gianfranco De Pascale, Linda Ejim, et al. The comprehensive antibiotic resistance database. *Antimicrobial agents and chemotherapy*, 57(7):3348–3357, 2013.

[8] Kortine Annina Kleinheinz, Katrine Grimstrup Joensen, and Mette Voldby Larsen. Applying the resfinder and virulencefinder web-services for easy identification of acquired antibiotic resistance and e. coli virulence genes in bacteriophage and prophage nucleotide sequences. *Bacteriophage*, 4(2):e27943, 2014.

[9] Torsten Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069, 2014.

[10] Martin Christen Frølund Thomsen, Johanne Ahrenfeldt, Jose Luis Bellod Cisneros, Vanessa Jurtz, Mette Voldby Larsen, Henrik Hasman, Frank Møller Aarestrup, and Ole Lund. A bacterial analysis platform: an integrated system for analysing bacterial whole genome sequencing data for clinical diagnostics and surveillance. *PloS one*, 11(6):e0157718, 2016.

[11] Alice R Wattam, David Abraham, Oral Dalay, Terry L Disz, Timothy Driscoll, Joseph L Gabbard, Joseph J Gillespie, Roger Gough, Deborah Hix, Ronald Kenyon, et al. Patric, the bacterial bioinformatics database and analysis resource. *Nucleic acids research*, 42(D1):D581–D591, 2013.

[12] Bala Swaminathan, Timothy J Barrett, Susan B Hunter, Robert V Tauxe, and CDC PulseNet Task Force. Pulsenet: the molecular subtyping network for foodborne bacterial disease surveillance, united states. *Emerging infectious diseases*, 7(3):382, 2001.

[13] Dennis A. Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and Eric W. Sayers. Genbank. *Nucleic Acids Research*, 41(D1):D36–D42, 2013.

[14] Matthew D Whiteside, Chad R Laing, Akiff Manji, Peter Kruczkiewicz, Eduardo N Taboada, and Victor PJ Gannon. Superphy: predictive genomics for the bacterial pathogen escherichia coli. *BMC microbiology*, 16(1):65, 2016.

[15] Matthew D Whiteside, Chad R Laing, and Victor PJ Gannon. Phylotyper: in silico predictor of gene subtypes. *Bioinformatics*, 2017.

[16] Chad Laing, Cody Buchanan, Eduardo N Taboada, Yongxiang Zhang, Andrew Kropinski, Andre Villegas, James E Thomas, and Victor PJ Gannon. Pan-genome sequence analysis using panseq: an online tool for the rapid analysis of core and accessory genomic regions. *BMC bioinformatics*, 11(1):461, 2010.

[17] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[18] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[19] Ian Horrocks, Bijan Parsia, Peter Patel-Schneider, and James Hendler. Semantic web architecture: Stack or two towers? In *International Workshop on Principles and Practice of Semantic Web Reasoning*, pages 37–41. Springer, 2005.

[20] Emma Griffiths, Damion Dooley, Morag Graham, Gary Van Domselaar, Fiona SL Brinkman, and William WL Hsiao. Context is everything: Harmonization of critical food microbiology descriptors and metadata for improved food safety and surveillance. *Frontiers in Microbiology*, 8:1068, 2017.

[21] Jerven T Bolleman, Christopher J Mungall, Francesco Strozzi, Joachim Baran, Michel Dumontier, Raoul JP Bonnal, Robert Buels, Robert Hoehndorf, Takatomo Fujisawa, Toshiaki Katayama, et al. Faldo: a semantic standard for describing the location of nucleotide and protein feature annotation. *Journal of biomedical semantics*, 7(1):39, 2016.

[22] Cátia Vaz, Alexandre P Francisco, Mickael Silva, Keith A Jolley, James E Bray, Hannes Pouseele, Joerg Rothganger, Mário Ramirez, and João A Carriço. Typon: the microbial typing ontology. *Journal of biomedical semantics*, 5(1):43, 2014.

[23] Wes Felter, Alexandre Ferreira, Ram Rajamony, and Juan Rubio. An updated performance comparison of virtual machines and linux containers. In *Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On*, pages 171–172. IEEE, 2015.

[24] Michael Inouye, Harriet Dashnow, Lesley-Ann Raven, Mark B Schultz, Bernard J Pope, Takehiro Tomita, Justin Zobel, and Kathryn E Holt. Srst2: Rapid genomic surveillance for public health and hospital microbiology labs. *Genome medicine*, 6(11):90, 2014.

[25] Samia N Naccache, Scot Federman, Narayanan Veeraraghavan, Matei Zaharia, Deanna Lee, Erik Samayoa, Jerome Bouquet, Alexander L Greninger, Ka-Cheung Luk, Barryett Enge, et al. A cloud-compatible bioinformatics pipeline for ultrarapid pathogen identification from next-generation sequencing of clinical samples. *Genome research*, 24(7):1180–1192, 2014.

[26] Alexandre G de Brevern, Jean-Philippe Meyniel, Cécile Fairhead, Cécile Neuvéglise, and Alain Malpertuy. Trends in it innovation to build a next generation bioinformatics solution to manage and analyse biological big data produced by ngs technologies. *BioMed research international*, 2015, 2015.

[27] Jeremy Goecks, Anton Nekrutenko, and James Taylor. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11(8):R86, 2010.

[28] Michael C Schatz. Biological data sciences in genome research. *Genome research*, 25(10):1417–1422, 2015.

[29] Rajeswari Swaminathan, Yungui Huang, Soheil Moosavinasab, Ronald Buckley, Christopher W Bartlett, and Simon M Lin. A review on genomics apis. *Computational and structural biotechnology journal*, 14:8–15, 2016.

8 APPENDIX