# Searching the Sequence Read Archive using Jetstream and Wrangler

Kyle Levi
San Diego State University - Biological and Medical Informatics Program
San Diego, California, United States of America
klevi@sdsu.edu

Mats Rynge
USC Information Sciences Institute
Marina del Rey, California, United States of America
rynge@isi.edu

Eroma Abeysinghe
Indiana University - Science Gateways Research Center
Bloomington, Indiana, United States of America
eabeysin@iu.edu

Robert A. Edwards
San Diego State University - Department of Computer Science
San Diego, California, United States of America
redwards@sdsu.edu

## ABSTRACT

The Sequence Read Archive (SRA), the world's largest database of sequences, hosts approximately 10 petabases ($10^{16}$ bp) of sequence data and is growing at the alarming rate of 10 TB per day. Yet this rich trove of data is inaccessible to most researchers: searching through the SRA requires large storage and computing facilities that are beyond the capacity of most laboratories. Enabling scientists to analyze existing sequence data will provide insight into ecology, medicine, and industrial applications. In this project we specifically focus on metagenomic sequences (whole community data sets from different environments). We are developing a set of tools to enable biologists to mine the metagenomes in the SRA using the NSF-funded cloud computing resources, Jetstream and Wrangler. We have developed a proof-of-principle pipeline to demonstrate the feasibility of the approach. We are leveraging our existing infrastructure to enable all scientists to access the SRA metagenomes regardless of their computational ability and are working to create a stable pipeline with a science gateway portal that is accessible to all researchers.

## CCS CONCEPTS

• **Applied computing** → **Bioinformatics**; *Molecular sequence analysis*; *Computational genomics*;

## KEYWORDS

Sequence Read Archive, SRA, Metagenomics, Jetstream, Wrangler, Bacteriophage, Apache Airavata, SciGaP, Credential Store, Search SRA, SRA Gateway, Metagenomics Discovery Challenge

**ACM Reference Format:**
Kyle Levi, Mats Rynge, Eroma Abeysinghe, and Robert A. Edwards . 2018. Searching the Sequence Read Archive using Jetstream and Wrangler. In *PEARC '18: Practice and Experience in Advanced Research Computing, July 22–26, 2018, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3219104.3229278



*PEARC '18, July 22–26, 2018, Pittsburgh, PA, USA*

ACM ISBN 978-1-4503-6446-1/18/07.
https://doi.org/10.1145/3219104.3229278

## 1 INTRODUCTION

A rapid drop in the cost of sequencing DNA - from roughly $10,000 dollars per Mbp in October of 2001 to less than $0.01 today - has fueled the rapid growth of the SRA from $10^{10}$ bases in March 2007 to $10^{16}$ today with no sign of slowing [4] This mass of data is the result of an international collaboration between the DNA Data Bank of Japan, European Bioinformatics Institute, and National Center for Biotechnology Information [5]. Despite creating the world's largest database of raw sequencing data, efforts to analyze the data have lagged far behind its growth, leaving a trove of unanalyzed biological data and the opportunity for big data experiments that would be preventatively expensive even with lower sequencing costs.

Data in the SRA is organized into studies, each of which contains one or more samples. Each sample has one or more experiments, and each experiment has one or more runs. At the time of writing there were 2,445,782 experiments and 2,776,555 runs in the SRA (approximately 80% of experiments have an only a single run). The genomics community established this database to enable sharing of the data, but the computational barrier to searching this data leaves the it separated from the people most qualified to analyze it.

Though there are many types of genomic data within the SRA, this project focuses on metagenomic datasets because these contain many different organisms and can attract a wider interest of questions compared to single organism runs. There are two popular metagenomics approaches: First, Amplicon sequencing where a single piece of DNA (e.g. the 16S gene from bacteria or COX1 gene from eukaryotes) are amplified from the mixed DNA of many organisms, and sequencing that en masse. These studies provide a taxonomic profile of an environment and are less computationally demanding, but only provide information about which organisms are present and are incapable of detecting any viruses that may be in the sample. The second type of study, whole shotgun (WGS) metagenomics, is where random samples of the genomes in the environment are sequenced, usually without amplification, and results in a mixture of all the DNA in the sample [10]. The analysis of those samples is computationally intensive but provides a detailed

view of the organisms in the environment and the biochemistry being performed by those organisms [10]. These studies give a holistic understanding of bacterial and viral communities never before possible.

Over the last decade, metagenomics sequencing has focused on understanding the role of microbes in the environment and reconstructing genomes out of environmental sequences. With the growth of the SRA, we can begin to approach metagenomics in a new way. Instead of asking what genomes or metabolisms are found in a particular environment, we can ask what environments contain a gene, protein, genome or metabolism of interest using the abundance of random sequences from diverse environments in the SRA. This massive volume of data can also be used to identify genes that are conserved across environments, or environments that are hotspots of microbial or gene diversification.

However, this kind of computational approach to microbial ecology requires large compute and storage capabilities, which are beyond the reach of most biologists. The WGS projects in the SRA are accumulating at roughly 3,000 runs per month (averaged from June 2016 to June 2017), and the combined WGS data sets exceeds 100TB of data [4].

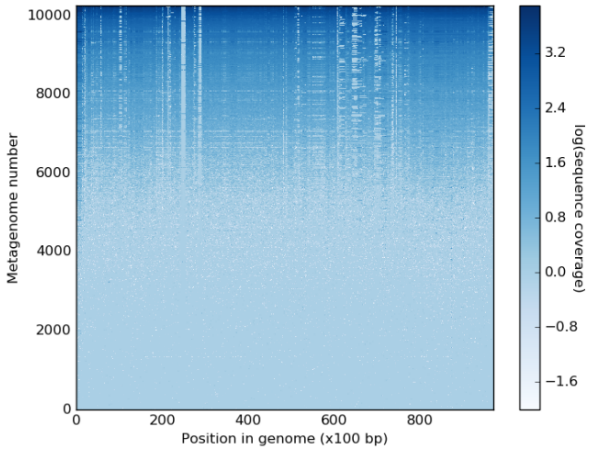


**Figure 1: Coverage of the crAssphage genome (x axis, position in bp) in 10,260 metagenomes (y axis). The coverage is based on log10(counts) as shown in the scale bar at right.**

## 2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

### 2.1 Searching the SRA Examples

*2.1.1 Investigating a newly discovered bacteriophage.* In 2014 Edwards and colleagues published the description of an 100 kb bacteriophage, a virus that infects Bacteroides, named crAssphage [9]. Previously this phage has been found in approximately half of the human intestinal metagenomics samples tested (n = 59). A heuristic approach was developed to expand this search and screen all WGS metagenome data sets within the SRA (screening 100,000 reads from each run) and used that approach to search the entire SRA for crAssphage. The full details of the approach can be found at https://github.com/linsalrob/SearchSRA.

The virus was found to be present in 10,260 runs as shown in Fig. 1. This figure demonstrates that some regions of the crAssphage genome are highly conserved (darker blue) while other regions are less well conserved (lighter blue). In particular, there are two genes that appear to be missing in most crAssphage genomes (the two lighter bands at 30kb). The presence/absence of these genes suggests a fundamental process in the evolution of this phage, which would not have been identified without the ability to investigate across many unique SRA datasets.

*2.1.2 Methane Cycle Proteins.* In a similar scan, the SRA was searched for two enzymes: particulate methane monooxygenase (PMO) and methyl-coenzyme M reductase (MCR) that are critical elements of the biological methane cycle [7]. RAPSearch2 [20] was used to compare 221 PMO and MCR protein sequences to the nucleotide sequences in the SRA. Using ten Jetstream [17][19] computes all SRA metagenomes were searched for the protein sequences in two days' time. As expected, many of the metagenomes did not have any similar sequences, but 9,149 metagenomes had at least one similarity with an expected value of $10^5$ or lower. Mapping those sequences against the PMO and MCR sequences identified variants of those enzymes. As evidence of the validity of this computational approach, the two runs with the highest number of hits were from samples where the investigators had specifically hunted MCR sequences by PCR (SRA runs SRR398144 and SRR2046417).

### 2.2 XSEDE Resources

Computational resources used for the searchers comes from two XSEDE resources: Jetstream [17] and Wrangler [11]. Both are co-located at both Indiana University (IU) and Texas Advanced Computing Center (TACC) but are two very different types of resources. The former is an OpenStack based compute cloud, while the latter is a data analysis system with a large flash based storage system. The resources where chosen because Jetstream can provide the elasticity required for a portal with a varying workload, VMs can be added automatically to meet the active user request, and Wrangler provides the required I/O to search the large amount of SRA data.

*2.2.1 Cloud Autoscaling with HTCondor.* Inside the Jetstream cluster, user search requests are handled by HTCondor [18], which is a high throughput computing system well suited for handling workloads of this type - large amount of single or small number of threads applications. When users submit a new search, the list of runs to be searched is checked against the SRA runs available on Wrangler. Missing SRAs are logged, while available SRA IDs are grouped together into HTCondor jobs. The jobs are added to a DAGMan workflow [8] (Fig. 2), with a top-level job indexing the reference genome, and a final local job to package up the results for final delivery to the user. The DAGMan workflow is submitted to the HTCondor queue on the service virtual machine [1][2].

To serve submitted searches, Jetstream virtual machines are auto-scaled based on the demand. The auto-scaler is implemented using OpenStack's shade library, which is a simple high-level Python module for interacting with OpenStack based clouds [3]. Every 5 minutes, a cron job checks for pending jobs in the HTCondor queue.

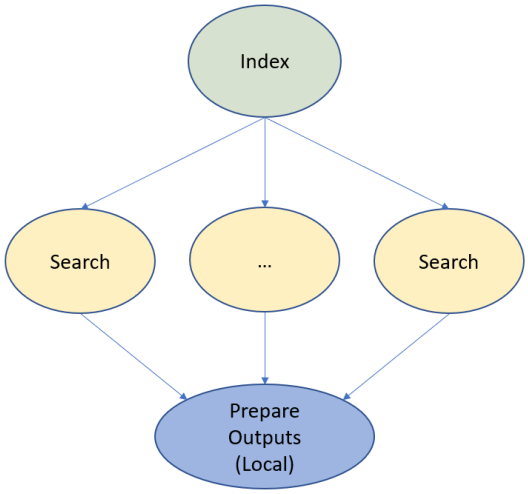


**Figure 2: Structure of the HTCondor DAGMan graph.**

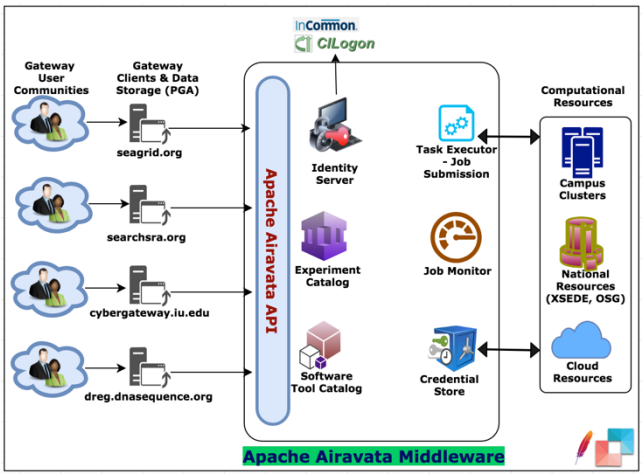


**Figure 3: Searching SRA Gateway, Apache Airavata Middleware & Computational Resources.**

A decision to scale up is based on 3 metrics: the number of existing virtual machines running jobs, the number of total pending jobs and the number of jobs that have been in queue for more than 15 minutes.

To quickly scale up for new searches, the auto-scaler is slightly more aggressive when starting from an empty cluster by only considering the total number of pending jobs. Once there are 3 or more virtual machines running, the auto-scaler switches to only considering the "old" jobs. The reason is to not oversubscribe - if jobs go into the queue and are served quickly, there's likely no need for additional resources. Currently, the total number of virtual machines are limited to 20, but the auto-scaler is under constant development and the scaling decision logic and limits will likely change as more users are added to the system.

*2.2.2 Wrangler Integration.* Metagenomes that are classified as WGS metagenomes (regardless of the environment from which they come) are mirrored to a local system and staged for comparisons. Each month, the incremental new data is downloaded to Wrangler and used to provide direct access to the data for Jetstream users.

In parallel with the mirroring, 100,000 reads used in the prescreening comparison are extracted and staged, integrating the new datasets into the existing pipeline. The entire data set will be saved for subsequent comparisons as required. Over the course of this project this process will be automated so that all data is automatically updated monthly. This automatic pipeline will be released in common workflow language, so others may automatically mirror components of the SRA.

The Wrangler directory is directly mounted by the Jetstream virtual machines using a dedicated OpenStack network and NFS. Each virtual machine has two network interfaces: one for the general communication between the virtual machines and the internet, and one for the special address space and route required for communication with Wrangler. The latter was configured by the Jetstream and Wrangler administrators. When booting the virtual machine and attaching the two networks, the default route and hostname came from the main network connection, and the default route provided by the Wrangler network is. This was accomplished by custom "enter" and "exit" scripts for the DHCP client on the virtual machines. For example, in the "enter" script, the default route is ignored based on the DHCP provided IP address. Both the auto-scaler and the search workflows codes are available on Github [1][2].

## 2.3 Science Gateway and Jetstream

*2.3.1 Searching SRA Gateway with Apache Airavata.* The Searching SRA science gateway uses Science Gateway Platform as a Service [16] (https://scigap.org/).

The SciGaP platform provides gateway services via Apache Airavata [13] middleware. The Searching SRA gateway requires user identity, accounts, authorization, and the ability to access XSEDE cloud computational resources Jetstream and Wrangler for computations and user data management as core features from hosted Apache Airavata middleware. Fig. 3 depicts the components of the Apache Airavata and its functional interactions with the gateway instance and computing resource. The hosted Apache Airavata (SciGaP platform) used for the Search SRA gateway is multi-tenanted and manages multiple science gateways.

*2.3.2 User Accounts & Gateway Access.* Newly created Searching SRA gateway user accounts require administrator approval for the user to access Searching SRA software. When user creates an account, they are in an "access pending" state; once the gateway administrator approves the account user will become a "gateway-user" who can launch Searching SRA jobs. The gateway's roles also support users with administrative privileges; restricted administrative privileges allow users to have read-only access to administrator views [14]. All these gateway users except for ones in pending state can submit jobs on XSEDE Jetstream cluster. The admin user has the authorization to control metadata for accessing Jetstream and running the Searching SRA application, to manage users, and to monitor and access all user experiments information. For authentication and authorization searching SRA gateway uses

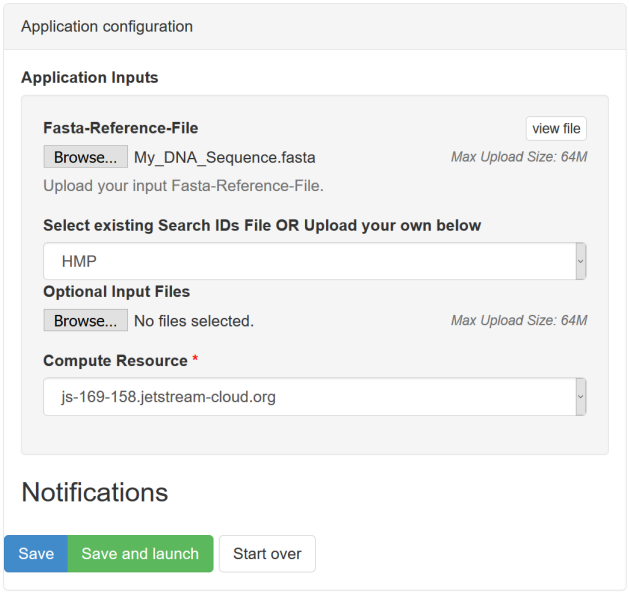


**Figure 4: Interface and required fields for submitting a scan.**

https://www.keycloak.org/ [6], the open source identity and access management solution.

*2.3.3 Searching Against SRA.* The searching SRA gateway users and gateway administrators can create, execute, monitor, share, and manage computational experiments. Experiments are created in the gateway to submit search jobs in to the Jetstream cluster. Fig. 4 depicts the interface for experiment creation. Using this interface, gateway users can upload search IDs or select search IDs file available in Jetstream to submit searching jobs into Jetstream. In order to make the interface user-friendly the interface is made simple, and users are only required to provide the required data files. The gateway decides where the computation runs as well as the properties in terms of nodes, CPUs and wall-time required for the computation. Once the search against the SRA is completed, users can download output data directly from Jetstream through the gateway. Users also have the option of sharing their work with other gateway users. As part of managing their experiments, users can cancel running experiments and clone existing and execute new experiments on Jetstream. Experiments can be searched using "Experiment Browse" interfaces, and searches can be filtered by creation date, application, experiment name, and description. Experiments are grouped in to Projects. Projects are shareable with other users similarly to experiments [15].

*2.3.4 Monitoring Job Progress.* Once an experiment is created and launched in the gateway, and the corresponding job is submitted to Jetstream, both the owner of the experiment (gateway user) and any gateway administrator can monitor the status. The experiment status can be monitored in two ways. One option is for the gateway users to provide their email address during experiment creation to receive messages at job start and end. The other option is to view the status in the "Experiment Summary" interface (Fig. 5), once the experiment is launched, the experiment summary

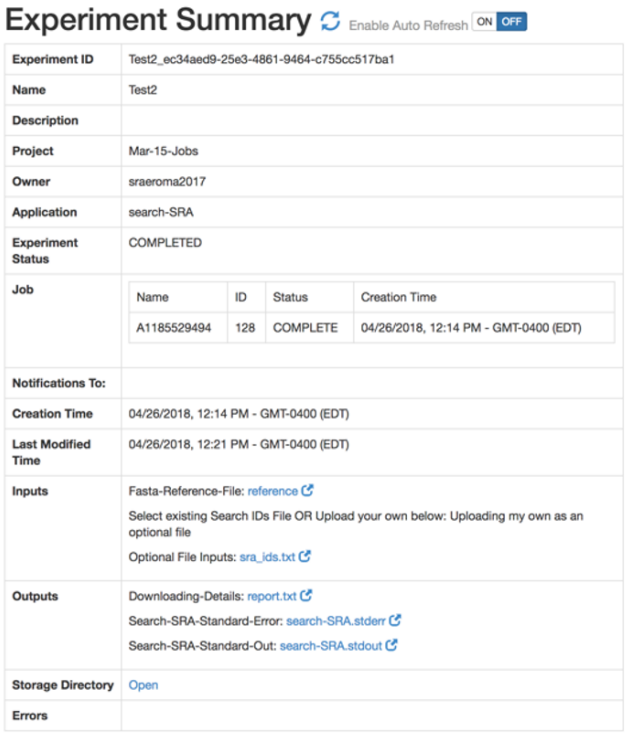


**Figure 5: Experiment Summary fields shown to users.**

interface is automatically refreshed to show the real-time status of the job submitted into the Jetstream. Regular users can monitor experiments owned by them and shared with them by other gateway users. Gateway administrators can monitor all gateway experiments using the Experiment Statistics page (Fig. 6) in the Admin Dashboard. This interface allows the gateway administrator to view the status of all experiments and job submissions.

*2.3.5 Gateway Administration.* The Admin Dashboard is the workspace for the gateway administrators. All the administrator features mentioned earlier are available through the Admin Dashboard. Apart from what is already discussed, the dashboard provides a notification feature, extensive user interfaces for managing gateway configurations required for compute resources and storage resources connectivity, and tools for managing credentials through Credential Store [12] for secure compute resource communications. Gateway notifications are for messages related to gateway operations, application availability and for and news related to Jetstream and Wrangler. In the Searching SRA gateway, administrators need to configure information required to connect to Jetstream to submit search jobs. These configurations include adding Jetstream login name, scratch location, preferred job submission protocol, and allocation project number. Similarly, Credential Store is used to generate an SSH credential token and key pair to be used in compute resource and storage resource communications.

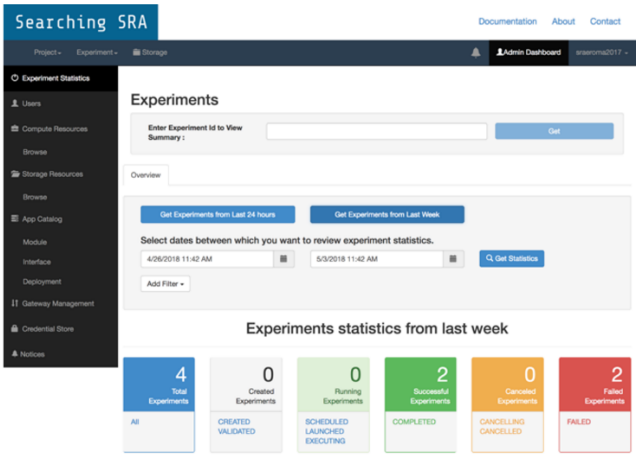


**Figure 6: Dashboard for Admin Users.**

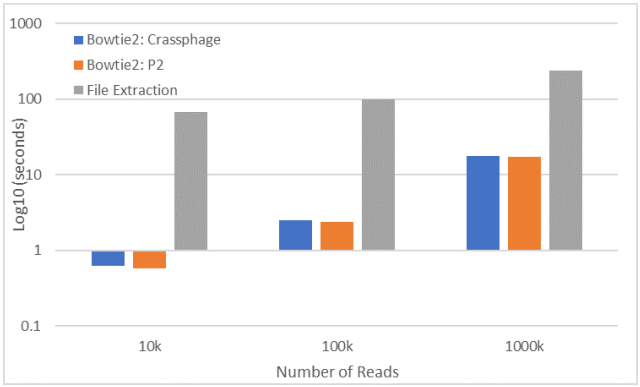


**Figure 7: Average time spent on each task for 87,702 SRA metagenomes sampled for 10,000, 100,000, and 1,000,000 reads.**

## 3 RESULTS AND DISCUSSION

### 3.1 Timing Results

*3.1.1 Extracting vs scanning metagenomes.* Without the high capacity storage provided by Wrangler, extracting SRA datasets becomes too time consuming to sample thousands or even hundreds of data sets. When runs are downloaded from the SRA, they are initially in a sparse file format (.sra) and must be extracted to the more common FASTQ format before they can be searched. Ignoring the time required to download the data sets (which can exceed 100 TB), the majority of the time spent analysing each run is extracting the data sets into the FASTQ format. Fig. 7 compares the average time spent per run on 3 different tasks: extracting the FASTQ file, searching for crAssphage with Bowtie2, and searching for P2 phage with Bowtie2. Scans for two different phages were included to compare the effect of genome size on Bowtie2 search time. Despite crAssphage having a genome almost three times as large as P2 (97kb vs 33kb), the search times differed by less than 2%.

File extraction is the most time expensive task by a large margin, consuming 95%, 92%, and 87% of the total time per run for 10,000, 100,000 and 1,000,000 reads respectively, when searching for two organisms. In a storage limited approach, these data sets are download, extracted, scanned and deleted in a process that can take days to weeks and offers no reusability. By storing the extracted data sets on Wrangler, search times are reduced to a fraction of that total time - in addition to saving time by being shared between researchers. All datasets have been pre-processed to eliminate the need for on-demand extraction of the sequences.

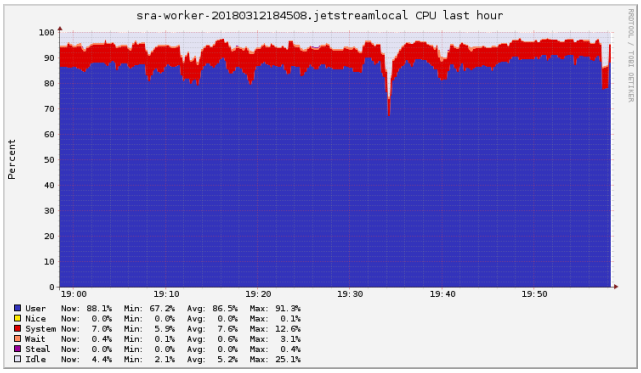


**Figure 8: Typical CPU usage for worker VM during a search.**

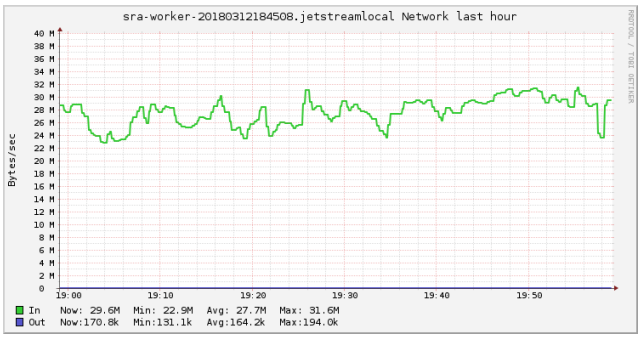


**Figure 9: Typical read rate for worker VM during a search.**

*3.1.2 SRA Gateway Searches.* Fig. 8 and Fig. 9 show typical CPU metrics and the network I/O for a worker VM during a search. In this instance, 86.5% of the CPU is busy in user space (running Bowtie2) and 7.6% is System space (mostly waiting for data from the Wrangler filesystem).

The network graph shows the consistent read rate with an average of 27.7 megabytes per second. These numbers, per virtual machine, stay consistent as the auto scaling is adding and removing virtual machines. We have not yet identified a bottleneck in the number of virtual machines we can scale to - the scaling is currently mostly constrained by the allocation and the default VM limits per user in Jetstream.

### 3.2 Heuristic Error

SRA datasets can range in size from megabytes to tens of gigabytes and it is a common occurrence for a file to contain few or no reads that map to the organism(s) being searched for. For this reason, 100,000 reads are sampled from each dataset to determine

if a sufficient quantity of an organism is present. This sampling introduces a source of false negatives (not finding an organism in a sample where it is present). The probability of a false negative is proportional to the total number of reads in the data set and inversely proportional to the percentage of reads belonging to the organism. This means the data sets that least contain the search organism are most likely to be false negatives. In this setting, false positives are impossible given the stringent matching requirements of Bowtie 2.

### 3.3 Training Future Bioinformaticians

Bioinformatics is a rapidly expanding field, and there is a strong need to train the next generation of researchers and industry professionals, but most classes lack the resources to introduce students to modern bioinformatic techniques. Keeping with the increasing demand, San Diego State University recently introduced a course aimed at teaching bioinformatics techniques to undergraduate biology students using the freely available data from the SRA, a generous allocation from Jetstream. The first iteration of the 16-week course included 7 weeks of training with Unix and programs related to searching the SRA, 3 weeks of training on available NCBI resources and 6 weeks to conduct an experiment related to antibiotic resistance or viruses. Future versions of this course aim to reduce the amount of Unix and BASH related training biology students must undergo and increase the time spent on data analysis and interpretation - a goal that will benefit immensely from the SRA Gateway.

### 3.4 Conclusion

The SRA Gateway, while still a work in progress, has already begun demonstrating its usefulness to students and researchers alike by solving two of the largest challenges when working with the SRA. First, the web interface removes the need for users to be experienced with Unix systems and commands before accessing SRA data and allows researchers from all backgrounds access to efficient parallel computing infrastructure to conduct experiments without requiring the explicit knowledge of the infrastructure itself. This interface also extends to students the opportunity to explore bioinformatics by conducting novel experiments with real world data. Second, by having the SRA datasets downloaded, extracted, preprocessed and hosted on Wrangler, search times are reduced by over 99%. This advantage saves the time of researches as well as computing resources by implementing efficient job scheduling with HTCondor and Jetstream instance autoscaling based on user demand. The final challenge when working with the SRA is data analysis. Presently, results from SRA searches are returned to users as a downloadable compressed folder of Binary Alignment Map (BAM) files, however, there are plans to introduce general data analysis to the pipeline using Python. The SRA is constantly accumulating new data and this vastly underutilized resource holds information relevant to many diverse areas of biology and medicine. It is the goal of the SRA Gateway to begin analyzing this plethora of data by removing the computational barriers between researches and the information contained in the SRA.

## ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

Development of the Apache Airavata used to develop the science gateway is supported by NSF award #1339774.

XSEDE resources used include JetStream and Wrangler at Indiana University through allocation TG-MCB170036. We thank Eroma Abeysinghe and Mats Rynge for their assistance with gateway and system integration, which was made possible through the XSEDE Extended Collaborative Support Service (ECSS).

## REFERENCES

[1] 2018. jetstream-search-sra: Search the SRA using the jetstream sra cluster setup. https://github.com/linsalrob/jetstream-search-sra original-date: 2018-01-05T15:56:45Z.

[2] 2018. jetstream-sra-cluster-setup: This module sets up an autoscaled Jetstream cluster for SRA work, based on HTCondor, the OpenStack Shade library, and SaltStack. https://github.com/linsalrob/jetstream-sra-cluster-setup original-date: 2018-01-05T15:52:04Z.

[3] 2018. shade: Client library for OpenStack containing Infra business logic. https://github.com/openstack-infra/shade original-date: 2015-01-07T21:07:08Z.

[4] 2018. SRA Documentation. https://www.ncbi.nlm.nih.gov/sra/docs/sragrowth/

[5] Jamie Alnasir and Hugh P. Shanahan. 2015. Investigation into the annotation of protocol sequencing steps in the sequence read archive. *GigaScience* 4, 1 (09 May 2015), 23. https://doi.org/10.1186/s13742-015-0064-7

[6] Marcus A. Christie, Anuj Bhandar, Supun Nakandala, Suresh Marru, Eroma Abeysinghe, Sudhakar Pamidighantam, and Marlon E. Pierce. 2017. Using Keycloak for Gateway Authentication and Authorization. (2017). https://doi.org/10.6084/m9.figshare.5483557.v1

[7] Ralf Conrad. 2009. The global methane cycle: recent advances in understanding the microbial processes involved. *Environmental Microbiology Reports* 1, 5 (2009), 285–292. https://doi.org/10.1111/j.1758-2229.2009.00038.x

[8] Peter Couvares, Tevfik Kosar, Alain Roy, Jeff Weber, and Kent Wenger. 2007. *Workflow Management in Condor.* Springer London, London, 357–375. https://doi.org/10.1007/978-1-84628-757-2_22

[9] Bas E. Dutilh, Noriko Cassman, Katelyn McNair, Savannah E. Sanchez, Genivaldo G. Z. Silva, Lance Boling, Jeremy J. Barr, Daan R. Speth, Victor Seguritan, Ramy K. Aziz, Ben Felts, Elizabeth A. Dinsdale, John L. Mokili, and Robert A. Edwards. 2014. A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes. *Nature Communications* 5 (2014), 4498. https://doi.org/10.1038/ncomms5498

[10] Robert A. Edwards and Forest Rohwer. 2005. Viral metagenomics. *Nature Reviews Microbiology* 3, 6 (2005), 504–510. https://doi.org/10.1038/nrmicro1163

[11] N. Gaffney, C. Jordan, T. Minyard, and D. Stanzione. 2014. Building Wrangler: A transformational data intensive resource for the open science community. In *2014 IEEE International Conference on Big Data (Big Data).* 20–22. https://doi.org/10.1109/BigData.2014.7004480

[12] T. A. Kanewala, S. Marru, J. Basney, and M. Pierce. 2014. A Credential Store for Multi-tenant Science Gateways. In *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (2014-05). 445–454. https://doi.org/10.1109/CCGrid.2014.95

[13] Suresh Marru, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh, Thilina Gunarathne, Eran Chinthaka, Ross Gardler, Aleksander Slominski, Ate Douma, Srinath Perera, and Sanjiva Weerawarana. 2011. Apache Airavata: A Framework for Distributed Applications and Computational Workflows. In *Proceedings of the 2011 ACM Workshop on Gateway Computing Environments* (2011) *(GCE '11).* ACM, 21–28. https://doi.org/10.1145/2110486.2110490

[14] S. Nakandala, H. Gunasinghe, S. Marru, and M. Pierce. 2016. Apache Airavata security manager: Authentication and authorization implementations for a multi-tenant escience framework. In *2016 IEEE 12th International Conference on e-Science (e-Science)* (2016-10). 287–292. https://doi.org/10.1109/eScience.2016.7870911

[15] Supun Nakandala, Suresh Marru, Marlon Piece, Sudhakar Pamidighantam, Kenneth Yoshimoto, Terri Schwartz, Subhashini Sivagnanam, Amit Majumdar, and Mark A. Miller. 2017. Apache Airavata Sharing Service: A Tool for Enabling User Collaboration in Science Gateways. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact* (2017) *(PEARC17).* ACM, 20:1–20:8. https://doi.org/10.1145/3093338.3093359

[16] Marlon Pierce, Suresh Marru, Borries Demeler, Amitava Majumdar, and Mark Miller. 2013. Science Gateway Operational Sustainability: Adopting a Platform-as-a-Service Approach. https://doi.org/10.6084/m9.figshare.790760.v1

[17] Craig A. Stewart, George Turner, Matthew Vaughn, Niall I. Gaffney, Timothy M. Cockerill, Ian Foster, David Hancock, Nirav Merchant, Edwin Skidmore, Daniel Stanzione, James Taylor, and Steven Tuecke. 2015. Jetstream: a self-provisioned, scalable science and engineering cloud environment. ACM Press, 1–8. https://doi.org/10.1145/2792745.2792774

[18] Douglas Thain, Todd Tannenbaum, and Miron Livny. 2005. Distributed computing in practice: the Condor experience. *Concurrency and Computation: Practice and Experience* 17, 2 (2005), 323–356. https://doi.org/10.1002/cpe.938

[19] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Computing in Science Engineering* 16, 5 (Sep 2014), 62–74. https://doi.org/10.1109/MCSE.2014.80

[20] Yongan Zhao, Haixu Tang, and Yuzhen Ye. 2012. RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics (Oxford, England)* 28, 1 (2012), 125–126. https://doi.org/10.1093/bioinformatics/btr595