SOFTWARE TOOL ARTICLE

Jupyter notebook-based tools for building structured datasets from the Sequence Read Archive [version 1; peer review: 2 approved with reservations]

Matthew N. Bernstein¹, Ariella Gladstein², Khun Zaw Latt³, Emily Clough⁴, Ben Busby⁴, Allissa Dillman⁴

¹Morgridge Institute for Research, Madison, Wisconsin, 53715, USA
²Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, 27599, USA
³Kidney Disease Branch, National Institute of Diabetes and Digestive and Kidney Diseases, Bethesda, Maryland, 20892, USA
⁴National Center for Biotechnology Information NLM, Bethesda, Maryland, 20894, USA

Abstract

The Sequence Read Archive (SRA) is a large public repository that stores raw next-generation sequencing data from thousands of diverse scientific investigations. Despite its promise, reuse and re-analysis of SRA data has been challenged by the heterogeneity and poor quality of the metadata that describe its biological samples. Recently, the MetaSRA project standardized these metadata by annotating each sample with terms from biomedical ontologies. In this work, we present a pair of Jupyter notebook-based tools that utilize the MetaSRA for building structured datasets from the SRA in order to facilitate secondary analyses of the SRA’s human RNA-seq data. The first tool, called the Case-Control Finder, finds suitable case and control samples for a given disease or condition where the cases and controls are matched by tissue or cell type. The second tool, called the Series Finder, finds ordered sets of samples for the purpose of addressing biological questions pertaining to changes over a numerical property such as time. These tools were the result of a three-day-long NCBI Codeathon in March 2019 held at the University of North Carolina at Chapel Hill.

Keywords

Hackathon, RNA-seq, Sequence Read Archive, MetaSRA, Metadata, Ontology, Jupyter

This article is included in the Hackathons collection.
Corresponding author: Matthew N. Bernstein (mbernstein@morgridge.org)

Author roles: Bernstein MN: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Gladstein A: Conceptualization, Formal Analysis, Software, Validation, Visualization; Latt KZ: Conceptualization, Formal Analysis, Investigation, Software, Visualization; Clough E: Conceptualization, Formal Analysis, Investigation, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Busby B: Funding Acquisition, Project Administration, Resources, Supervision; Dillman A: Funding Acquisition, Project Administration, Resources, Supervision

Competing interests: No competing interests were disclosed.

Grant information: This work was supported in part by the Intramural Research Program of the National Library of Medicine, National Institutes of Health. Matthew Bernstein acknowledges support from grant 2018-182626 from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2020 Bernstein MN et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Bernstein MN, Gladstein A, Latt KZ et al. Jupyter notebook-based tools for building structured datasets from the Sequence Read Archive [version 1; peer review: 2 approved with reservations] F1000Research 2020, 9:376 https://doi.org/10.12688/f1000research.23180.1

First published: 19 May 2020, 9:376 https://doi.org/10.12688/f1000research.23180.1
Introduction
The Sequence Read Archive (SRA; Leinonen et al., 2011) is a large public repository that stores next-generation sequencing data from thousands of diverse scientific investigations. Despite its promise, reuse and re-analysis of SRA data has been challenged by the heterogeneity and poor quality of the metadata that describe its biological samples (Gonçalves & Musen, 2019). Recently, the MetaSRA project (Bernstein et al., 2017) standardized these metadata by annotating each sample with terms from biomedical ontologies including Cell Ontology (Bard et al., 2005), Uberon (Mungall et al., 2012), Disease Ontology (Schriml et al., 2019), Cellosaurus (Bairoch, 2018), and the Experimental Factors Ontology (Malone et al., 2010). The MetaSRA also features an interface (http://metasra.biostat.wisc.edu) for querying human RNA-seq samples using these ontology term annotations. However, the MetaSRA web interface is not capable of producing structured datasets such as those that match case samples associated with a target condition or disease with healthy control samples. Similarly, the MetaSRA is also not capable of searching for samples associated with a particular condition and/or tissue-type that are ordered according to a numeric property (e.g., age).

Construction of such datasets is non-trivial and requires further processing of the results provided by the MetaSRA website. For example, finding case and control samples for a given disease likely requires matching case samples to control samples according to their tissue or cell type. Furthermore, given these search results, users may wish to further filter samples according to whether they are poorly annotated (i.e., are missing cell type or tissue information), whether they are derived from a cell line, or whether they were experimentally treated. Moreover, given these results, the user may wish to explore other ontology terms associated with the search results within either the case or control samples to check for any variables that may confound downstream analyses. Finding longitudinal or time-series data presents similar challenges. To the best of our knowledge, no existing tool addresses these tasks.

To address these two tasks, we produced two Jupyter notebook-based tools. The first tool, called the Case-Control Finder, searches the SRA via the MetaSRA terms to produce matched-case and control samples for a given disease or condition where the order is determined based on a temporal property in the metadata such as age. The second tool, called the Series Finder, finds ordered sets of samples that are associated with a numerical property (e.g., time point) for a given tissue or cell type. More specifically, the Series Finder produces ordered sets of samples, where the order is determined based on a temporal property in the metadata such as age, as standardized by the MetaSRA’s real-valued properties. These tools promise to facilitate the construction of suitable public datasets for secondary analyses.

Methods
The tools presented in this work were written in Python (v3.6) and make use of Python packages pandas (McKinney, 2011), Matplotlib (Hunter, 2007), and seaborn (https://seaborn.pydata.org). These notebooks are available ready-to-run in a Docker container.

Case-Control Finder
The Case-Control Finder implements the following steps to produce a dataset of matched-case control samples for a given disease (Figure 1A):

1. Generate candidate case and control samples. Generate the set of candidate case samples by querying for all samples associated with a user-specified condition or disease using the MetaSRA-mapped ontology terms. Also, find all candidate control samples that are not associated with the target condition/disease.

2. Filter poorly annotated samples. Filter samples based on a metadata completeness threshold, which requires that all samples be associated with either a tissue term or a cell type term. The tissue/cell type information is required for downstream matching of case samples to control samples.

3. Apply user-specified filters. Further filter samples according to user-specified filtering parameters. The user can filter out cell line samples, treated samples, and in vitro differentiated samples. The user can also remove all diseased samples from the candidate control samples for the purpose of generating a healthy control-set.

4. Match by tissue and cell type. The candidate case samples are then matched with the candidate control samples by their tissue and cell type terms. Specifically, given that each sample can be associated with multiple ontology terms in the MetaSRA, a set of case samples is matched with a set of control samples when both sets of samples are labelled with the same set of tissue and cell type terms. For example, a set of case samples annotated with the set of terms “liver” and “epithelial cell” will be matched only to control samples also labeled strictly with these terms (Figure 2A). This ensures that case samples are matched with maximally similar control samples and mitigates matching samples from different tissue-types. For example, a set of case samples labelled with both the terms “liver” and “epithelial cell” will not be matched with a set of samples labelled only as “epithelial cell,” as there is no guarantee that the latter set of samples originate in the liver.

Once the dataset is constructed, the notebook enables the user to explore the samples for other MetaSRA mapped ontology terms within the data (Figure 2B and C). By presenting other common ontology terms in the data, the user may be able to identify variables that either confound analysis.

Series Finder
The Series Finder finds RNA-seq data samples that are associated with a numerical property (e.g., age or time point) for a given tissue or cell type. To do so, the Series Finder utilizes the real-value property annotations provided by the MetaSRA.
Figure 1. Data flows for hypothesis-driven query tools. An overview of the backend processing functions called from the Jupyter notebooks.

Figure 2. Example results from the Case-Control Finder. Results from running the Case-Control Finder for the query “liver cancer.” (A) The Case-Control Finder displays the number of case/control studies (left) and case/control samples (right) matched by each tissue and cell type. (B) The user can select either the case samples or control samples for a given tissue or cell type and display the most common ontology terms associated with those selected samples. Displayed here are the most common terms associated with the case samples labeled as “liver.” (C) The notebook also displays four pie charts for viewing the fraction of samples belonging to a cell line (top left), each sex (top right), each developmental stage (bottom left), and whether they were given an experimental treatment (bottom right).
where each real-value property in the MetaSRA is structured as a tuple consisting of a property name (e.g., age), numerical value, and unit (e.g., year).

To perform a query, the user provides an ontology term, such as a tissue or cell type, as well as a property name and unit. The Series Finder then finds all samples that are associated with the target ontology term and real-value property. The user can also provide a set of blacklist terms that can be used to filter the samples. Given a list of blacklist terms, the Series Finder will remove all samples annotated with any blacklist term. The Series Finder will then return all remaining samples ordered by their associated numerical values (Figure 1B).

**Results and use cases**

We used the Case-Control Finder to query for samples of liver cancer RNA-seq samples matched with healthy control samples. This query resulted in six sets of samples representing different tissues or cell types including epithelial cells, hepatocytes, stem cells, and liver tissue (Figure 2A). The Case-Control Finder identified common terms associated with the case “liver cancer” samples (Figure 2B), and categorized these samples by cell line status, sex, developmental stage, and treatment status (Figure 2C).

We used the Series Finder to find all brain samples in the SRA ordered by the age of the sample donor. This query resulted in samples spanning many ages (Figure 3A). This dataset could prove useful for exploring gene expression-based signatures of aging. The Series Finder also identified common terms at each age (Figure 3B) and for each age’s sample-set, categorized those samples by cell line status, sex, developmental stage, and treatment status (Figure 3C).

**Conclusion and future work**

We implemented two Jupyter notebooks for performing hypothesis-driven queries of public RNA-seq samples in the SRA. These tools are built upon the standardized metadata provided by the MetaSRA project and enable querying of the metadata beyond what is natively possible via the MetaSRA website interface. Future work will entail either integrating these tools into a standard web-interface, such as the interface of...
the MetaSRA website, or by implementing a stand-alone web application for these tools using a platform such as R Shiny.

Data availability
The figures and datasets produced in the analyses can be found on GitHub: https://github.com/mbernste/hypothesis-driven-SRA-queries/tree/master/results

Software availability
All code is maintained on GitHub: https://github.com/mbernste/hypothesis-driven-SRA-queries

Acknowledgements
We would like to thank Carl Leubsdorf and Brad Plecs for technical support using Google Cloud Platform servers, and J. Rodney Brister and Barton Trawick for administrative support.

References


Archived code as at time of publication: https://doi.org/10.5281/zenodo.3807512 (Bernstein, 2020)

License: CC0
Open Peer Review

Current Peer Review Status: ?

Version 1

Reviewer Report 05 June 2020

https://doi.org/10.5256/f1000research.25586.r63614

© 2020 Ellis S. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Shannon Ellis

1 Department of Biostatistics, Johns Hopkins University School of Public Health, Baltimore, MD, USA
2 Department of Cognitive Science, UC San Diego, La Jolla, CA, USA

This paper describes the development of two Jupyter notebook-based tools (Case-Control Finder and Series Finder) for improving the ease with which researchers can identify cases within the SRA for further study.

While the paper does a nice job describing what the tool is and how it can be helpful and the code & examples provided/explained the paper function as expected (is reproducible), there are a few limitations in its implementation that will limit its utility with researchers:

1. The fact that this tool requires a static version of the SRA metadata to be loaded in limits its ability to be updated and requires the authors to manually download the metadata - access by API to SRA would improve this process.

2. While the provided examples work well, there are limitations to unfamiliar users and failures in cases that seem on reading the paper like they should work.
   - For example: in series finder if I change `term` to "heart" (instead of "brain"), almost all subsequent cells fail.
   - In case-control finder, if I change `condition` to "brain cancer", all but one samples returned are controls (which does not align with what is in the SRA?) and visualization formatting becomes difficult.
   - By clarifying what user options are (or examples) for each place where user is free to play with the input, this could be avoided. Similarly, functions lack documentation and examples here or checks on input within the functions, so diving into the code becomes critical for use, which will limit users. Adding documentation and checks for user input could assist in this overall.

Minor issues:
1. I was able to download locally using the "not recommended" approach; however, docker asked for a password using suggested approach in README (I didn't investigate further).
2. In the paper & notebooks, tool would be improved by focusing on readability of visualizations. For example, flipping the bar charts in figure 2A by 90 degrees (and accompanying in the notebook), the labels would be more readable. And, by considering the colors in figure C, such that "orange" is not used in all three pie charts (when they do not represent the same categories) would be helpful. Having the number of samples summarized by the pie charts would also be helpful.

3. The sentence in Introduction starting with "More specifically, the Series Finder produces..." is unclear. Specifically, on reading, I'm not sure what a temporal property would be in the metadata (other than the listed age). As a reader, this limits my understanding of 1 of the two notebooks provided and my ability to use the tool.

4. I may be missing it, but it seems like cases and controls would benefit most from being able to also be matched on age and sex to truly make them useful for further analysis. It does not seem this functionality exists, or I'm missing it.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: genetics, bioinformatics, data science education

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 01 June 2020
https://doi.org/10.5256/f1000research.25586.r63613

© 2020 Wang Z. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Bernstein et al. provides two Jupyter notebook-based tools to facilitate re-analysis of human RNA-seq data deposited to SRA. The tools were built on top of annotated metadata of RNA-seq samples from the MetaRNA, and provided some visualizations of the summary statistics of the query results.

I have the following suggestions and comments:

1. The authors should indicate how to access the Jupyter notebooks in the abstract.

2. It would require less overhead for users if the authors make their Jupyter notebook tools available to execute on Binder or Google Colab.

3. Since MetaSRA mapped RNA-seq samples to biomedical ontologies, it would be useful to have the Jupyter tools also enable query using ontology terms in addition to free texts. For instance, a researcher may want to focus on samples from non-small cell lung carcinoma (DOID:3908) rather than any types of lung cancers.

4. Currently, both notebooks load the metadata of the SRA samples from a preprocessed file in the Git repository. It would be useful to make it interoperable with MetaSRA through API to be able to query against the most updated version of SRA, which may include many more samples. As the volume of public RNA-seq data are drastically increasing.

5. Please provide available options for the structured query, including "target_property" and "UNIT", in the "Series Finder" notebook.

6. Please provide assessment of the precision and recall of the tools in terms of retrieving the correct samples given queries.

7. Can the authors please comment on the applicability of the tools on bulk vs. single-cell samples?

8. Please add discussion about how to perform secondary analysis on the SRA samples after obtaining the structured data from the Jupyter notebooks.

Is the rationale for developing the new software tool clearly explained?  
Yes

Is the description of the software tool technically sound?  
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?  
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?  
Partly
Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** Bioinformatics; Computational Biology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.