SOFTWARE TOOL ARTICLE

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

Matthew N. Bernstein[1], Ariella Gladstein[2], Khun Zaw Latt[3], Emily Clough[4], Ben Busby[4], Allissa Dillman[4]

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

v2 **First published:** 19 May 2020, **9**:376
https://doi.org/10.12688/f1000research.23180.1
Latest published: 04 Aug 2020, **9**:376
https://doi.org/10.12688/f1000research.23180.2

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.

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	**2**
version 2 (revision) 04 Aug 2020	✓ report	✓ report
version 1 19 May 2020	? report	? report

1. **Zichen Wang**, Sema4, Stamford, USA

2. **Shannon Ellis**, UC San Diego, La Jolla, USA
Johns Hopkins University School of Public Health, Baltimore, USA

Any reports and responses or comments on the article can be found at the end of the article.

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.

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 2; peer review: 2 approved]** F1000Research 2020, **9**:376
https://doi.org/10.12688/f1000research.23180.2

First published: 19 May 2020, **9**:376 https://doi.org/10.12688/f1000research.23180.1

REVISED **Amendments from Version 1**

In this revision, we present a number of significant updates to these tools. First, in order to facilitate the use of these notebooks, we have made them available to run in the cloud via Google Colab. Second, the queries now utilize the ontology graph structure to return samples that are annotated as an ancestral term to the query term according to the ontology graphs and thus, the queries return more results. Third, in the Case-Control Finder, we have implemented the ability to match cases to controls by sex and age in addition to tissue and cell type. Fourth, we've coalesced many of the notebook cells in order to make the tools simpler and easier to use. Fifth, users can formulate queries using ontology term ID's (e.g. "DOID:3571", the term ID for "liver cancer"). Sixth and finally, we fixed Figure 2C. In the previous version of the manuscript this figure displayed data for the incorrect subset of patients.

Any further responses from the reviewers can be found at the end of the article

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. Specifically, finding case and control samples for a given disease requires matching case samples to control samples according to their tissue or cell type. For example, if one were to naively search the MetaSRA for "liver cancer" samples, the results would include samples from Kim *et al.* (2020), which consist of isolated T cells from liver tumors. Therefore, only matched T cell samples would make for appropriate controls. 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 cases and controls are matched by tissue and cell type. The second tool, called the *Series Finder*, finds ordered sets of samples for the purpose of answering biological questions pertaining to changes over a numerical property (e.g., time). More specifically, the Series Finder produces ordered sets of samples, where the order is determined based on a temporal property in the metadata as standardized by the MetaSRA's real-valued properties. Examples of temporal properties include the age of a person from which a given sample originated or the time in which a given sample of cells have spent differentiating *in vitro*. 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 can be run in the cloud via Google Colab. A link to these notebooks can be found in the README within the Github repository (https://github.com/mbernste/hypothesis-driven-SRA-queries).

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, cell type, age, and sex.** The candidate case samples are then matched with the candidate control samples by their tissue and cell type terms. Optionally, the user can also match samples by

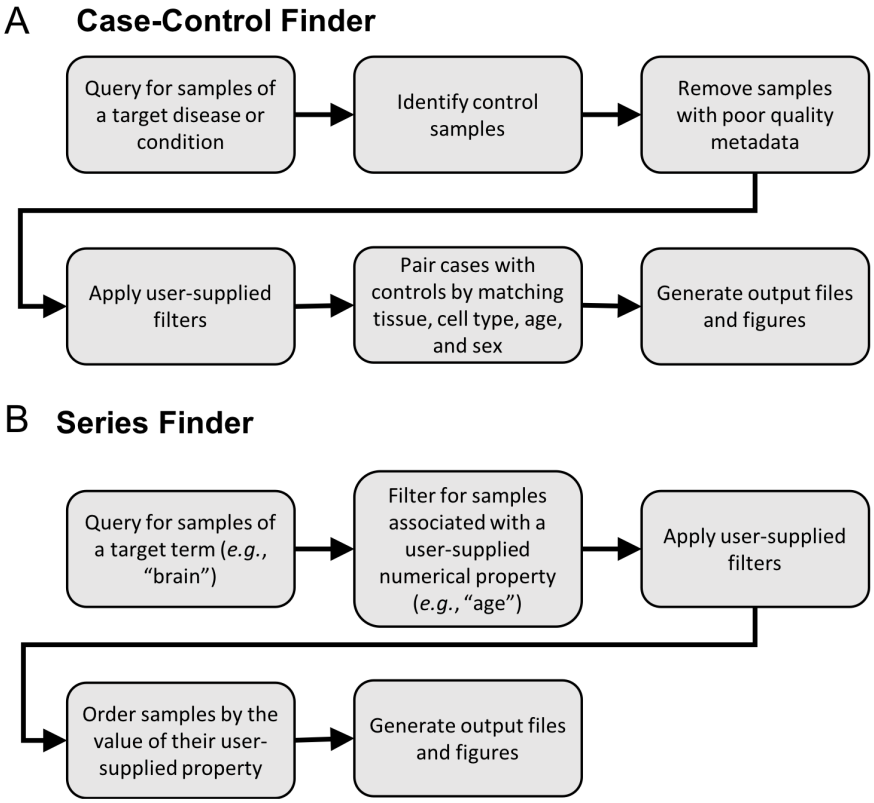

Figure 1. Data flows for hypothesis-driven query tools. An overview of the backend processing functions called from the Jupyter notebooks.

age and sex. Specifically, given that each sample can be associated with ultiple 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 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 specify a set of filters (e.g. for filtering diseased samples or cell line samples) and the Series Finder will remove all samples that meet the filter specification. 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 21 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).

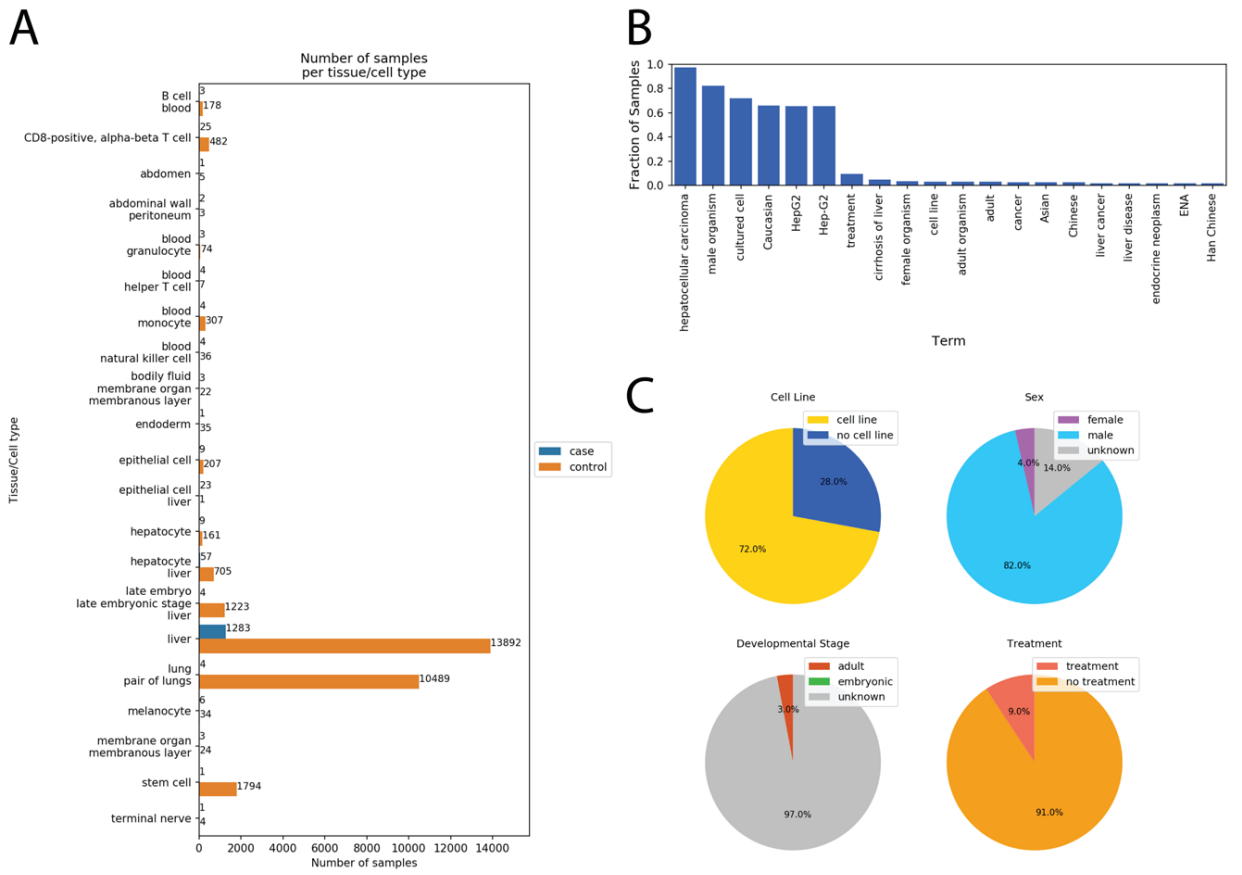

Figure 2. Example results from the Case-Control Finder. Results from running the Case-Control Finder for the query "liver cancer." (**A**) The Case-Control Finder displays the number of case/control samples matched by each tissue and cell type. (**B**) The user can select either the case samples or control 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).

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. Given the SRA accessions of the RNA-seq samples that these tools produce, a user can then retrieve the gene expression data for these samples in order to perform secondary analyses. Specifically, the user can either download and process the raw reads from the SRA, or they can obtain preprocessed gene expression profiles from recent mass preprocessing efforts such as recount2 (Collado-Torres, 2017), ARCHS4 (Lachmann *et al.*, 2018), and refine.bio (Greene *et al.*). Finally, these notebooks come pre-packaged with metadata files from the latest version of the SRA, as provided by the SRAdb (Yuelin *et al.*, 2013), and MetaSRA. When the MetaSRA releases a new version of annotated metadata, these notebooks will be updated to track the new release.

We also note a few limitations to this work. First given that the MetaSRA annotates the SRA samples using an automated computational pipeline, its annotations contain some errors. Errors in the MetaSRA may propagate to the results produced by these tools, and thus, the datasets produced by these tools are best utilized as sets of *candidate* datasets for downstream analysis. We point the reader to Bernstein *et al.* (2017) for an analysis of the MetaSRA's accuracy. We also note that the SRA stores sequencing data for both bulk RNA-seq and single-cell RNA-seq samples; however, this information is not encoded in any standardized way within the SRA nor is it captured by the

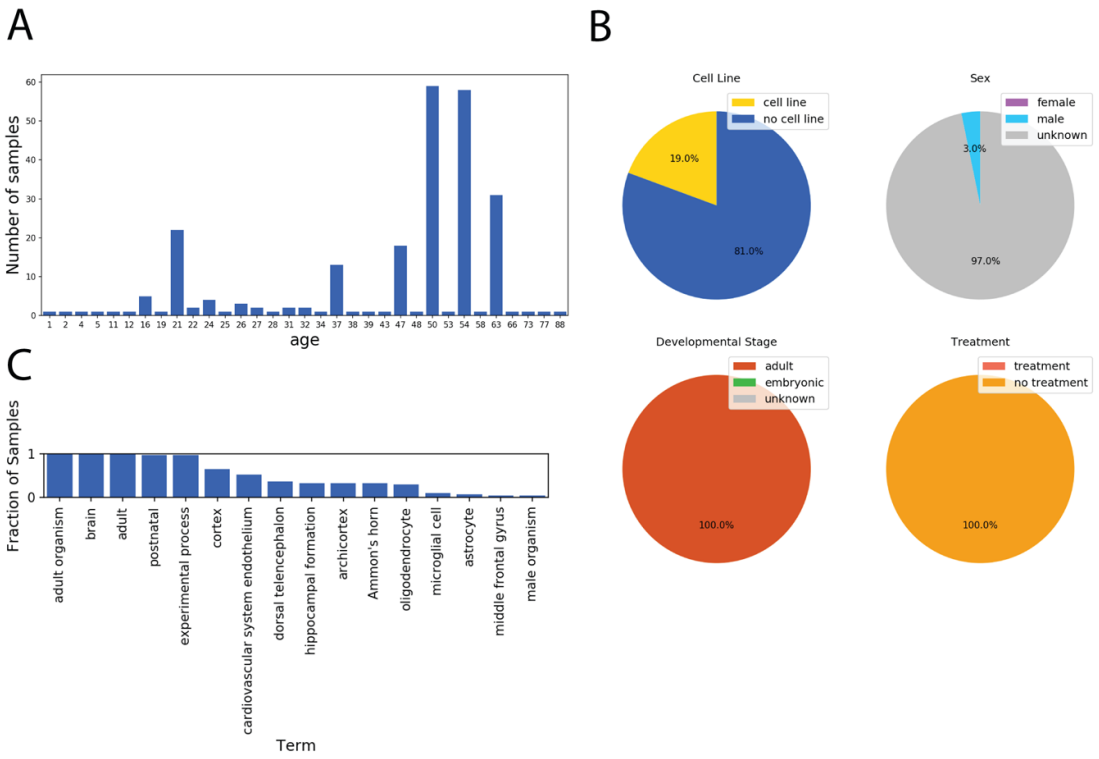

Figure 3. Example results from the Series Finder. Results from running the Series Finder for the query "brain" sorted by "age," where unit is specified as "year." (**A**) The Series Finder displays the number of samples sorted by age. (**B**) The user can select samples associated with a given time point for further exploration. Here the samples annotated as "year = 63" are selected. The notebook then 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). (**C**) Given the selected samples from (**B**), the notebook displays the most frequent terms associated with those selected samples.

MetaSRA. Thus, results returned by these tools may include a mixture of both single-cell and bulk data. For these reasons, we encourage users to validate the results returned by these tools by consulting their entries in the SRA before proceeding with downstream analyses. Lastly, to facilitate access to these tools, it would benefit to implement them within an easy-to-use web interface rather than Jupyter notebooks. Future work will entail either integrating these tools into the MetaSRA website, or 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

Archived code as at time of publication: https://doi.org/10.5281/zenodo.3957949 (Bernstein, 2020)

License: CC0

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

Bairoch A: **The Cellosaurus, a Cell-Line Knowledge Resource.** *J Biomol Tech.* 2018; **29**(2): 25–38.
PubMed Abstract | Publisher Full Text | Free Full Text

Bard J, Rhee SY, Ashburner M: **An ontology for cell types.** *Genome Biol.* 2005; **6**(2): R21.
PubMed Abstract | Publisher Full Text | Free Full Text

Bernstein M: **mbernste/hypothesis-driven-SRA-queries: First release (Version v1.0.0).** *Zenodo.* 2020.
http://www.doi.org/10.5281/zenodo.3807512

Bernstein MN, Doan A, Dewey CN: **MetaSRA: normalized human sample-specific metadata for the Sequence Read Archive.** *Bioinformatics.* 2017; **33**(18): 2914–2923.
PubMed Abstract | Publisher Full Text | Free Full Text

Collado-Torres L, Nellore A, Kammers K, *et al.*: **Reproducible RNA-seq analysis**

using *recount2*. *Nat Biotechnol.* 2017; **35**(4): 319–321.
PubMed Abstract | Publisher Full Text | Free Full Text

Gonçalves RS, Musen MA: **The variable quality of metadata about biological samples used in biomedical experiments.** *Sci Data.* 2019; **6**: 190021.
PubMed Abstract | Publisher Full Text | Free Full Text

Greene CS, Hu D, Jones RWW, *et al.*: **refine.bio: a resource of uniformly processed publicly available gene expression datasets**.
Reference Source

Hunter JD: **Matplotlib: A 2D graphics environment.** *Comput Sci Eng.* 2007; **9**(3): 90–95.
Publisher Full Text

Kim H, Park S, Jeong S, *et al.*: **4-1BB Delineates Distinct Activation Status of Exhausted Tumor-Infiltrating CD8$^+$ T Cells in Hepatocellular Carcinoma.** *Hepatology.* 2020; **71**(3): 955–971.
PubMed Abstract | Publisher Full Text | Free Full Text

Lachmann A, Torre D, Keenan AB, *et al.*: **Massive mining of publicly available RNA-seq data from human and mouse.** *Nat Commun.* 2018; **9**(1): 1366.
PubMed Abstract | Publisher Full Text | Free Full Text

Leinonen R, Sugawara H, Shumway M, *et al.*: **The Sequence Read Archive.** *Nucleic Acids Res.* 2011; **39**(Database issue): D19–21.
PubMed Abstract | Publisher Full Text | Free Full Text

Malone J, Holloway E, Adamusiak T, *et al.*: **Modeling sample variables with an Experimental Factor Ontology.** *Bioinformatics.* 2010; **26**(8): 1112–1118.
PubMed Abstract | Publisher Full Text | Free Full Text

McKinney W: **pandas: a foundational Python library for data analysis and statistics.** *Python for High Performance and Scientific Computing.* 2011; **14**.
Reference Source

Mungall CJ, Torniai C, Gkoutos GV, *et al.*: **Uberon, an integrative multi-species anatomy ontology.** *Genome Biol.* 2012; **13**(1): R5.
PubMed Abstract | Publisher Full Text | Free Full Text

Schriml LM, Mitraka E, Munro J, *et al.*: **Human Disease Ontology 2018 update: classification, content and workflow expansion.** *Nucleic Acids Res.* 2019; **47**(D1): D955–D962.
PubMed Abstract | Publisher Full Text | Free Full Text

Yuelin Z, Stephens RM, Meltzer PS, *et al.*: **SRAdb: query and use public next-generation sequencing data from within R.** *BMC Bioinformatics.* 2013; **14**: 19.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 24 August 2020

https://doi.org/10.5256/f1000research.27241.r68621

✔ **Zichen Wang**
Sema4, Stamford, CT, USA

The authors have addressed all of my comments and significantly improved the tools.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics; Computational Biology; Machine Learning

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.

Reviewer Report 12 August 2020

https://doi.org/10.5256/f1000research.27241.r68620

✔ **Shannon Ellis**
[1] Department of Cognitive Science, UC San Diego, La Jolla, CA, USA
[2] Department of Biostatistics, Johns Hopkins University School of Public Health, Baltimore, MD, USA

Thank you to the authors for their thoughtful response and changes to the manuscript & notebooks. I'm happy with the current version of the manuscript and associated materials.

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.

Version 1

Reviewer Report 05 June 2020

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

?

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

Author Response 23 Jul 2020

Matthew Bernstein, Morgridge Institute for Research, Madison, USA

We greatly appreciate the reviewer's valuable suggestions and feedback. Please see our responses below:

1. We agree that using the MetaSRA's API would be a great idea; however, the API restricts queries that return too many results. Specifically, for queries that return too many results, the API returns an error message that the search results are too large. This severely restricts our ability to use the API for these tools. We note that the MetaSRA is released in discrete chunks and does not track every ongoing change to the SRA; thus, whenever the MetaSRA version changes, we will update the static version of the MetaSRA packaged with these tools. We have added text to this manuscript detailing our commitment to performing these updates. Lastly, we added text to the README that makes it more explicit to the user which version of the MetaSRA these tools are utilizing.

2.

- We tested the query "heart" and it now should return results. We also provide more thorough input validation for cases in which the query does not return results.

- We have updated the code so that the tools retrieves sample that are annotated as an ancestral term to the query term (e.g. a sample labelled as "brain glioma" should be retrieved when the user inputs the query "brain cancer"). Now the query "brain cancer" will retrieve many more samples than before. We do note a few issues with the particular query "brain cancer" (which maps to term DOID:1319 in the Disease Ontology). Specifically, we found that the MetaSRA failed to label many samples as "brain cancer" due to the fact that many of the subterms (e.g. "brain glioma") are missing important synonyms that would have led the MetaSRA to pick them up. For example, the term "brain glioma" (DOID:0060108) is not associated with the simple synonym "glioma" and thus, unless a sample for a given glioma sample was described using the string "brain glioma", which appears to be rare, the MetaSRA failed to annotate this sample as a "brain glioma". Instead, the MetaSRA labels glioma samples using an alternative "glioma" term from the Experimental Factors Ontology (EFO:0005543), which does not have "brain cancer" as an ancestor term, but instead has "brain neoplasm" as an ancestor (EFO:0003833). This case points to the fact that there is still work to be done in both standardizing the metadata in the SRA and in constructing comprehensive ontologies. Unfortunately, these issues remain out of the scope for this work; however, we now include new text in the Conclusion section that discusses how the original MetaSRA annotations contain some errors and that these errors may propagate to the output of these tools.

- Thank you for this suggestion. We have added more detailed instructions for each input parameter. We also perform more thorough input-validation on the user's input. Lastly, we have added more documentation to each function in utils to help a user who wishes to dive further into the code.

Responses to minor issues:

1. We apologize for this password issue. Given how few dependencies these notebooks utilize, we decided that Docker is probably overkill for this project and therefore we removed this option altogether. We instead uploaded these notebooks to Google Colab to run in the cloud. If a user would like to run the notebooks locally, we now detail all of the dependencies in the file "requirements.txt" within the repository and offer guidance on installing these dependencies in the README.

2. Thank you for these suggestions. We flipped the barcharts 90 degrees and also use a different color palette for each pie chart. We note that the same samples are used to construct each of the four pie charts.

3. We added text to this sentence highlighting another example of a temporal property: time in which cells have spent differentiating in vitro. To this end, we have also added another parameter to the query that enables users to select only in vitro differentiating cells in order to answer possible biological questions pertaining to differentiation.

4. This is definitely an important feature, thank you for suggesting it. We now enable the user to match by age and sex in the notebook (see Section "3. Set filtering parameters") in the notebook. Specifically, in the notebook, if the user sets the variable "MATCH_BY_SEX" to True, we only consider samples that are annotated by sex in the MetaSRA and then match accordingly. Similarly, if the user sets "MATCH_BY_AGE" to True, we only consider samples that are annotated with age and then match accordingly.

Competing Interests: No competing interests were disclosed.

Reviewer Report 01 June 2020

https://doi.org/10.5256/f1000research.25586.r63613

?

Zichen Wang
Sema4, Stamford, CT, USA

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.

Author Response 23 Jul 2020

Matthew Bernstein, Morgridge Institute for Research, Madison, USA

We greatly appreciate the reviewer's valuable feedback. Please find our responses to each point below:

1. Within the abstract we now point the reader to the tools' Github repository, which describes how the tools can be executed either locally or in the cloud via Google Colab.

2. We have set up Google Colab notebooks to run these tools in the cloud. Links to the notebooks are found within the README in the Github repository.

3. We thank you for this suggestion. We have updated the tools to now accept both ontology term names (i.e. free text) as well as ontology term ID's.

4. We agree that using the MetaSRA's API would be a great idea; however, the API restricts queries that return too many results. Specifically, for queries that return too many results, the API returns an error message that the search results are too large. This severely restricts our ability to use the API for these tools. We note that the MetaSRA is released in discrete chunks and does not track every ongoing change to the SRA; thus, whenever the MetaSRA version changes, we will update the static version of the MetaSRA packaged with these tools. We have added text to this manuscript detailing our commitment to performing these updates. Lastly, we added text to the README that makes it more explicit to the user which version of the MetaSRA these tools are utilizing.

5. Within the instructions (within Section 1 of the Series Finder), we now provide the user example properties (such as "passage number" and "time") as well as example units (such as "hour" and "day"). We also point the user to the Units Ontology for a full set of available units that are utilized by the underlying MetaSRA annotations.

6. We note that the accuracy of the results is dependent on the accuracy of the MetaSRA annotations, which have been thoroughly evaluated in the original MetaSRA publication by Bernstein et al. (2017). Therefore, we added text to the "Conclusion and future work" section that points readers to this analysis. We have also added text to this section that clarifies that these tools are for selecting an initial *candidate* set of samples from the SRA; however, given that the annotations are not error-free, we encourage the user to further validate the datasets returned by these tools before performing downstream analysis.

7. The SRA stores sequencing data for both bulk and single-cell data; however, this information is not encoded in the metadata in a standardized way nor is it captured by the MetaSRA. Therefore, one limitation of the tools presented in this work is that they may return datasets that comprise both bulk and single-cell samples. We describe this limitation

in the Conclusion section and again encourage users to validate the results returned by these tools before performing downstream analyses.

8. In the Conclusion section, we now point the reader to databases of pre-processed SRA data including recount2, ARCHS4, and refine.bio. From these resources, users can download pre-processed expression data for the samples returned by the tools presented in this work.

Competing Interests: No competing interests were disclosed.