SOFTWARE TOOL ARTICLE

REVISED Increasing workflow development speed and reproducibility with Vectools [version 2; peer review: 2 approved]

Tyler Weirick[1,2], Raphael Müller[2,3], Shizuka Uchida[1,2,4]

[1]Cardiovascular Innovation Institute, University of Louisville, Louisville, KY, 40202, USA
[2]Institute of Cardiovascular Regeneration, Goethe University Frankfurt, Frankfurt am Main, Hessen, 60590, Germany
[3]Institute for Bioinformatics and Systems Biology, Justus Liebig University Giessen, Giessen, Hessen, 35392, Germany
[4]Institute of Molecular Cardiology, University of Louisville, Louisville, KY, 40202, USA

v2 **First published:** 20 Sep 2018, **7**:1499
https://doi.org/10.12688/f1000research.16301.1
Latest published: 23 Oct 2018, **7**:1499
https://doi.org/10.12688/f1000research.16301.2

Abstract
Despite advances in bioinformatics, custom scripts remain a source of difficulty, slowing workflow development and hampering reproducibility. Here, we introduce Vectools, a command-line tool-suite to reduce reliance on custom scripts and improve reproducibility by offering a wide range of common easy-to-use functions for table and vector manipulation. Vectools also offers a number of vector related functions to speed up workflow development, such as simple machine learning and common statistics functions.

Keywords
bioinformatics, reproducibility, workflow, vector, matrix, spreadsheet

This article is included in the International Society for Computational Biology Community Journal gateway.

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	**2**
version 2 (revision) 23 Oct 2018	✓ report	✓ report
version 1 20 Sep 2018	? report	

1 **Yutaka Saito**, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

2 **Dimitar Vassilev**, Sofia University "St. Kliment Ohridski", Sofia, Bulgaria

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

Corresponding author: Shizuka Uchida (heart.lncrna@gmail.com)

Author roles: Weirick T: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Müller R**: Software, Writing – Review & Editing; **Uchida S**: Funding Acquisition, Project Administration, Resources, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This study was supported by the start-up funding from the Mansbach Family, the Gheens Foundation and other generous supporters at the University of Louisville; University of Louisville 21st Century University Initiative on Big Data in Medicine (Z1762); and the Deutsche Forschungsgemeinschaft (SFB834 Z4).
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: Weirick T, Müller R and Uchida S. **Increasing workflow development speed and reproducibility with Vectools [version 2; peer review: 2 approved]** F1000Research 2018, **7**:1499 https://doi.org/10.12688/f1000research.16301.2

First published: 20 Sep 2018, **7**:1499 https://doi.org/10.12688/f1000research.16301.1

REVISED **Amendments from Version 1**

In this revised version, we implemented two additional operations in Vectools, which are: 1) "mode" for calculating mode/antimode in Vectools; and 2) "colmerge" for combining or splitting columns based on a delimiter. We also added the "--group" option to relevant operations (e.g., mean, mode, stdev). Due to these changes, it is now possible to perform similar functions as Bedtools as indicated in the main text. We updated Figure 1 by adding the file sizes for the SVM example. Supplementary File 1 has also been updated.

See referee reports

Introduction

Although the importance of computational analyses in biological research is increasingly appreciated, many analyses are time consuming to implement and remain complicated, as well as being difficult to reproduce[1]. Workflow-managers [e.g., Snakemake[2]] have greatly simplified many aspects needed for reproducibility. However, custom scripts (i.e., software not intended for use by a wider audience) remain a problem, which hampers the increased shareability offered by workflow-managers[3]. Custom scripts are often needed to further process data generated by high-use programs (i.e., programs intended for a wide user base). At the most basic level, analysis pipelines requiring custom scripts simply take more time to implement as additional code needs to be written. In addition, writing custom scripts also increases the chance of software bugs, which is concerning as even small bugs have led to retractions, such as mislabeling metadata[4] or a sign change[5]. Furthermore, analyses using custom scripts also hamper reproducibility as the scripts may be publically unavailable, lack documentation, or does not work on certain operation systems. To reduce the impact of these problems, we introduce Vectools[6], a command-line tool for working with vectors, matrices, and tables. Vectools reduces the need for custom scripts by offering an easy-to-use command-line tool with a wide range functions for manipulating tables, one of the most commonly used formats in bioinformatics. Further, Vectools incorporates a number of other useful vector-related functions, such as statistics and machine learning. Altogether, Vectools helps to speed up workflow development and improves reproducibility by offering a wide range of useful functions.

Methods

Implementation

Vectools can be run via command-line by simply typing "vectools", which will print the main help menu. Vectools contains over 45 operations organized by headings. These are analysis, descriptors, manipulation, math, normalization, supervised learning, and unsupervised learning. A list of all headings and functions is available in (Supplementary File 1). To run an operation, simply type "vectools" followed by the operation name. If the "—help" argument is added after an operation name, a help menu with usage instructions and examples will be printed.

Operation

A standard laptop computer with a recent version of Python3 will handle most applications.

Use cases

When manipulating data in tables, Core Utilities (Coreutils) programs (e.g., awk, grep, sed, and join) can be used instead of custom scripts. Using Coreutils helps to solve problems with availability as they are common to Unix-based systems. Here, we compared the usage of Vectools to Coreutils. Methods and output can be found in the archived data[7]. One downside of Coreutils programs is that they can be complex and difficult to understand. For example, joining multiple tables requires a Bash script using Coreutils-join, whereas this can be done with a single line with Vectools (Figure 1A). Furthermore, while common in Unix systems, the behavior of Coreutils programs may differ depending on the operating system. These differences can potentially cause errors or unexpected behavior, such as aggregating Gene Ontology (GO) terms by gene accession numbers with sed (Figure 1B). Instead of aggregating values on MacOS or other Berkeley Software Distribution (BSD) Unix systems, the Coreutils function prints the original input data. These errors can be caused by multiple reasons, such as BSD-sed not interpreting ANSI-C escape sequences (e.g., \n for newline, \t for tab) and differences in how regular expressions are evaluated. These problems can be overcome with Vectools with only one line of command. Vectools offers many functions that are currently unavailable in Coreutils, such as basic machine learning. Here, we show a simple example of using a support-vector machine to find potential novel carbonic anhydrases independent of sequence homology (Figure 1C). Carbonic anhydrases were chosen as they have multiple distinct classes, which arose via convergent evolution[8]. Vectools significantly simplifies a number of steps needed for this task. For example, the "svmtrain" operation handles hyper-parameter tuning via grid search, k-fold testing, and independent set testing. This significantly simplifies implementing machine learning in analysis pipelines.

Discussion

Here, we show that Vectools reduces the need for custom scripts and is simpler to use than Coreutils. While Coreutils is faster and uses less memory, this is generally a minor issue given the increasing power and decreasing cost of computational resources. Although format-specific tools (e.g., Bedtools[9]) offer similar functionalities, the generalized design of Vectools allows the majority of these functionalities to be replicated by combining Vectools operations with pipes. Furthermore, Vectools includes various other functionalities not available in Coreutils or format-specific tools, such as allowing easy incorporation of machine learning into analysis pipelines. Users may also be interested in comparison with R. While certainly suited to the same tasks: 1) integrating R into a pipeline requires custom scripts; and 2) the use-cases for R and Vectools are different. R offers a large variety of functions at the cost of package dependency issues. Conversely, Vectools emphasizes ease-of-use

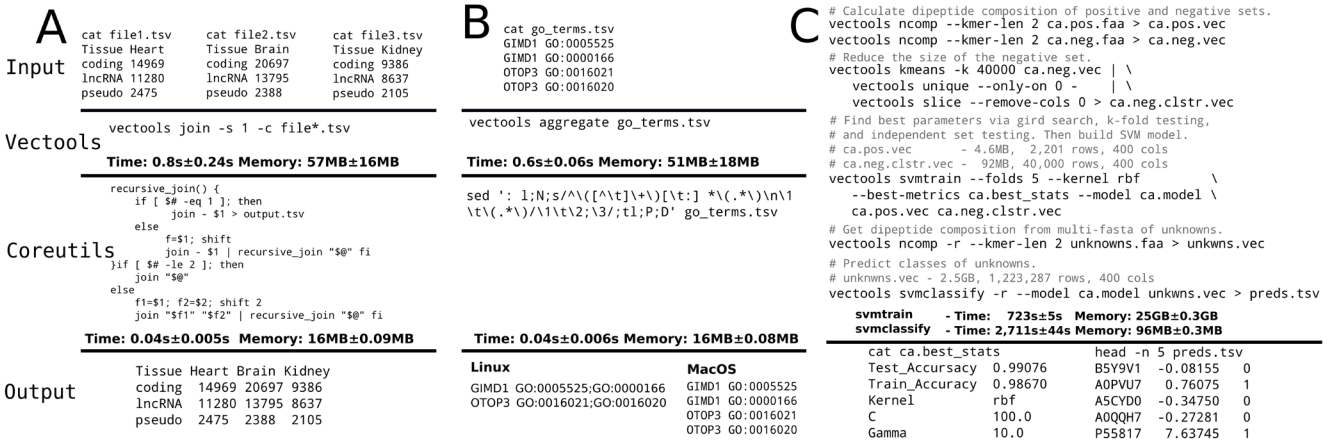

Figure 1. Comparison of Vectools and Coreutils. (**A**) Joining more than two files requires a single command using Vectools. The same operation using Coreutils requires a custom script. The information regarding file sizes is omitted as whole files are shown. (**B**) Aggregating Gene Ontology terms by gene accession numbers with Vectools can be done with a simple command. The same operation using Coreutils requires a complex regular expression. Further, the regular expression does not work properly on MacOS. The information regarding file sizes is omitted as whole files are shown. (**C**) Vectools also includes many operations unavailable in Coreutils, such as machine learning. Here, in five commands, we use supervised-learning for homology-independent prediction of enzyme function. Using Vectools we generated a support-vector machine model capable of predicting carbonic anhydrases with an estimated 99% accuracy and predict 15,018 of 1,223,287 uncharacterized proteins as potential carbonic anhydrases. The size and dimensions of files used in the machine learning examples are shown in the image as comments. Additionally, methods, input, and output data can be found in the archived data and analysis pipelines[7].

by hosting a curated list of common functions, helps to increase reproducibility by making analysis pipelines easier to share, and reduces bugs by omitting the need for custom scripts. Thus, one common use-case of Vectools when combined with a workflow-manager is to replace work done in spreadsheets. This use-case offers a number of benefits. For example, it is in line with a recent technology feature in *Nature*, which argues that the concept of reproducibility extends to creating easy-to-update analysis pipelines[10]. With Vectools, these easy-to-update pipelines will also be easy to share, making it a valuable tool for bioinformatics research.

Data availability

All data used in the paper are archived in Zenodo[7].

Software availability

Source code available from: https://vectools.bitbucket.io/.

Data and analysis pipelines: http://doi.org/10.5281/zenodo.1413666[7].

Source code at time of publication: http://doi.org/10.5281/zenodo.1413671[6].

Grant information

This study was supported by the start-up funding from the Mansbach Family, the Gheens Foundation and other generous supporters at the University of Louisville; University of Louisville 21st Century University Initiative on Big Data in Medicine (Z1762); and the Deutsche Forschungsgemeinschaft (SFB834 Z4).

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

Supplementary material

Supplementary File 1. A list of operations offered by Vectools with short descriptions of their functions
Click here to access the data

References

1. Fehr J, Heiland J, Himpe C, *et al.*: **Best practices for replicability, reproducibility and reusability of computer-based experiments exemplified by model reduction software.** *AIMS Mathematics.* 2016; **1**(3): 261–281.
Publisher Full Text
2. Köster J, Rahmann S: **Snakemake--a scalable bioinformatics workflow engine.** *Bioinformatics.* 2012; **28**(19): 2520–2522.
PubMed Abstract | Publisher Full Text
3. LeVeque RJ: **Top ten reasons to not share your code (and why you should anyway).** *SIAM News.* 2013; **1**.
Reference Source
4. Henson KE, Jagsi R, Cutter D, *et al.*: **Retraction.** *J Clin Oncol.* 2016; **34**(27): 3358–3359.
PubMed Abstract | Publisher Full Text
5. Ma C, Chang G: **Structure of the multidrug resistance efflux transporter EmrE from *Escherichia coli*.** *Proc Natl Acad Sci U S A.* 2007; **104**(9): 3668.
PubMed Abstract | Publisher Full Text | Free Full Text
6. Weirick T, Müller R, Uchida S: **Vectools source code at time of publication.** *Zenodo.* 2018.
http://www.doi.org/10.5281/zenodo.1413671
7. Weirick T, Müller R, Uchida S: **Data and analysis pipelines used in Increasing workflow development speed and reproducibility with Vectools [Data set].** *Zenodo.* 2018.
http://www.doi.org/110.5281/zenodo.1413666
8. Hewett-Emmett D, Tashian RE: **Functional diversity, conservation, and convergence in the evolution of the alpha-, beta-, and gamma-carbonic anhydrase gene families.** *Mol Phylogenet Evol.* 1996; **5**(1): 50–77.
PubMed Abstract | Publisher Full Text
9. Quinlan AR, Hall IM: **BEDTools: a flexible suite of utilities for comparing genomic features.** *Bioinformatics.* 2010; **26**(6): 814–2.
PubMed Abstract | Publisher Full Text | Free Full Text
10. Perkel JM: **A toolkit for data transparency takes shape.** *Nature.* 2018; **560**(7719): 513–515.
PubMed Abstract | Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 03 December 2018

https://doi.org/10.5256/f1000research.18308.r40346

✔ **Dimitar Vassilev**
Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”, Sofia, Bulgaria

The submitted material concerns a useful domain in applicative bioinformatics - development of pipelines and platforms for more or less general purposes and services in bioinformatics analyses. The major objective is to suspend the reliance of custom designed scripts and to improve the reproducibility of such software solutions.

I think that the authors have managed to a large extent the improvement of the reproducibility of their software’s set of tools for vector and table manipulation, including also some basic statistical inferences and machine learning models.

The efforts of the authors to have a formal general purpose platform at the end promised to have an obvious potential. Although, it is bold to speak about the certain large scale generality of the presented software – Vectools - because of the limited number of models incorporated.

The lack of some very new models in Vectools is an open question and I hope that the authors are able to support the openness of the software and its future improvement. In regards to the performance speed of Vectools - it will be good to include a sort of api concerning some valuable information for speed and memory usage (which is not an easily-predicted problem in machine learning models).

In regards to whether sufficient details of the code, methods and analysis were provided to allow replication of the software development and its use by others, I have answered ‘Partly’ to this question, because I think it would be worth providing the users of the software product with better documentation, which does not diminish the positive sides of the submitted material and the code.

In regards to whether sufficient information is provided to allow interpretation of the expected output datasets and any results generated using the tool, the interpretation of the expected outputs is specific to every case, and depends very much on the data so should not be always certain in the very proper discussion of the results. In this line I would advise the authors to show a couple of references with wider theoretical basis for better discussion and interpretation of the outcomes of the submitted software product.

In regards to whether the conclusions about the tool and its performance are adequately supported by the findings presented in the article, I think the conclusions serve to the submitted material very well but it would be worth showing the specificity of the software product in order to avoid the implied total generality of the work.

Finally, I support the submitted material to be indexed, with some amendments.

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?
Partly

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, Statistics, Data Science, 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 29 October 2018

https://doi.org/10.5256/f1000research.18308.r39811

Yutaka Saito
Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

In this revision, the authors have addressed all of my comments.
I think the article is now suitable for indexing.

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

Is the description of the software tool technically sound?
Partly

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

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

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 26 September 2018

https://doi.org/10.5256/f1000research.17808.r38645

?

Yutaka Saito
Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

This article describes Vectools, a command-line tool that can do various kinds of matrix operations for tsv-like data with simple one-liner programs. Vectools is similar to sed and awk commands in Unix Coreutils but has more functionalities, thereby reducing the cost for implementing custom scripts for daily data analyses. The authors claim this will improve the reproducibility problem in recent bioinformatics studies.

As a general comment, I think Vectools is useful and will be of interest for bioinformaticians who work in practical data analyses. Although I do not feel the tool has a theoretical novelty, its practical usefulness is worth post-publication evaluation by future users.

I have several comments as follows:

1:
- Vectools is also similar to "groupby" function in Bedtools.
- Some functionalities of Bedtools groupby are not included in Vectools, and vice versa.
- The authors should refer to Bedtools, and if any, other command-line tools similar to Vectools.

2:
- For each analysis in Figure 1, please provide the size of input data (#rows, #columns, #sequences, etc.).
- Especially, I get the impression that SVM consumes a large memory.
- Although I partly agree with the authors' statement that the computational cost is a minor issue, it is still important to provide the information of memory usage along with data size.

3 (minor points):
- (Top left in page 2) However --> In addition (?)
- (Top right in page 3) valuable tool --> valuable tools

References

1. Quinlan AR, Hall IM: BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*. 2010; **26** (6): 841-2 PubMed Abstract | Publisher Full Text

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

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?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics

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 19 Oct 2018

Shizuka Uchida, University of Louisville, Louisville, USA

We would like to thank the reviewer for valuable comments. The followings are our point-by-point responses:

> Comment #1: I have several comments as follows:

- **Vectools is also similar to "groupby" function in Bedtools.**
- **Some functionalities of Bedtools groupby are not included in Vectools, and vice versa.**
- **The authors should refer to Bedtools, and if any, other command-line tools similar to Vectools.**

> *Our response: Thank you very much for raising this point. We now clearly cite Bedtools in the Discussion section. To address the functionality issue, we have implemented two additional operations in Vectools, which are: 1) "mode" for calculating mode/antimode in Vectools; and 2) "colmerge" for combining or splitting columns based on a delimiter. We have also added the "--group" option to relevant operations (e.g., mean, mode, stdev). For cases in which the operation names or functionality do not match exactly, we list the equivalences between Bedtools Groupby and Vectools below:*

Groupby - Vectools
count– shape | slice
collapse– aggregate
distinct– unique
count_distinct– unique | sum
sstdev– vrep | stdev
freqasc/ freqdesc– unique | slice | colmerge | aggregate
first/ last– chop

> Comment #2:

- **For each analysis in Figure 1, please provide the size of input data (#rows, #columns, #sequences, etc.).**
- **Especially, I get the impression that SVM consumes a large memory.**
- **Although I partly agree with the authors' statement that the computational cost is a minor issue, it is still important to provide the information of memory usage along with data size.**

> *Our response: We have updated the figure by adding the file sizes for the SVM example. The first two examples display the entire file. Thus, we did not add file sizes in those examples. We have updated the figure legend to make this clearer. Further, all data used is assessable in the archived data. We have also updated the figure legend to make this more apparent. Finally, we fixed two typos in the figure.*

> Comment #3 (minor points):

- **(Top left in page 2) However --> In addition (?)**

- **(Top right in page 3) valuable tool --> valuable tools**

> *Our response: Thank you very much for reading our manuscript carefully. We have corrected the above grammatical errors as well as others.*

Competing Interests: No competing interests were disclosed.