


SOFTWARE

Open Access



Bioinformatics recipes: creating, executing and distributing reproducible data analysis workflows

Natay Aberra¹, Aswathy Sebastian¹, Aaron P. Maloy², Christopher B. Rees², Meredith L. Bartron² and Istvan Albert^{1*} 

* Correspondence: istvan.albert@gmail.com

¹Department of Biochemistry and Molecular Biology, Pennsylvania State University, 201 Old Main, University Park, PA 16802, USA
Full list of author information is available at the end of the article

Abstract

Background: Bioinformaticians collaborating with life scientists need software that allows them to involve their collaborators in the process of data analysis.

Results: We have developed a web application that allows researchers to publish and execute data analysis scripts. Within the platform bioinformaticians are able to deploy data analysis workflows (recipes) that their collaborators can execute via point and click interfaces. The results generated by the recipes are viewable via the web interface and consist of a snapshot of all the commands, printed messages and files that have been generated during the recipe run.

A demonstration version of our software is available at <https://www.bioinformatics.recipes/>. Detailed documentation for the software is available at: <https://bioinformatics.recipes.readthedocs.io>.

The source code for the software is distributed through GitHub at <https://github.com/ialbert/biostar-central>.

Conclusions: Our software platform supports collaborative interactions between bioinformaticians and life scientists. The software is presented via a web application that provides a high utility and user-friendly approach for conducting reproducible research. The recipes developed and shared through the web application are generic, with broad applicability and may be downloaded and executed on other computing platforms.

Keywords: Data analysis, Scientific workflows, Reproducibility



Background

The majority of bioinformatics analyses consist of several customizable computational tasks chained together to form a so-called pipeline or workflow. Publishing, documenting, and sharing these computational analyses are the cornerstones of reproducible research [1–4].

In this paper, we present a web application that allows bioinformaticians to publish and execute data analysis workflows. We call these workflows “bioinformatics recipes.” A “recipe” can be thought of as a standalone data analysis script that runs in a computing environment. A recipe may be a collection of several command-line tools, it may be a Makefile, a SnakeMake [3] file, a Nextflow [4] pipeline, an R script or any command line oriented computer program.

We designed our framework such that any series of commands may be formatted and published as a recipe. In addition, the application that we have developed can generate a graphical user interface to each recipe, thus facilitates user interaction and parameter selection at runtime.

Implementation

Our software is a Python and Django based application that can be installed and run with minimal system administration knowledge and is aimed to be deployed to serve individual research groups.

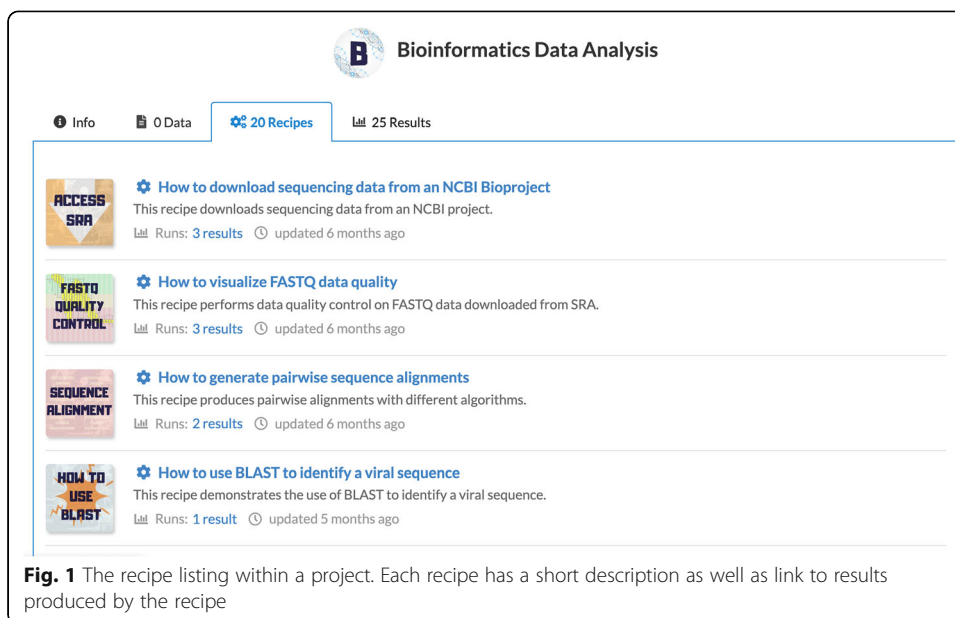
Our software also offers project-based laboratory data management. Within the management interface, all content is grouped into projects that may have public or private visibility. Content stored in public projects is readable without restrictions. Private projects will restrict access to members only. Within each project content is divided into three main categories:

1. Data (the input files)
2. Recipes (the code that processes the data)
3. Results (the directory that contains the resulting files of applying the recipe to data)

Figure 1 shows a project view with Data, Recipes and Results displayed in separate tabs of the project. A typical workflow requires that one or more Data are combined with a Recipe to produce a Result: Data + Recipe -> Results.

First the data section must be populated. Data may be uploaded or may be linked directly from a hard drive or from a mounted filesystem, thus avoiding copying and transferring large datasets over the web. For recipes that connect to the internet to download data, for example when downloading from the Short Read Archive the data does not need to be already present in the local server.

Notably the concept of “data” in our system is broader and more generic than that for a typical file system. In our software “data” may be a single file, it may be a compressed archive containing several files or it may be a path to a directory that contains any number of files as well as other subdirectories. The programming interfaces for recipes can handle directories transparently and make it possible to run the same recipes that one would use for a single file on all files of an entire directory.

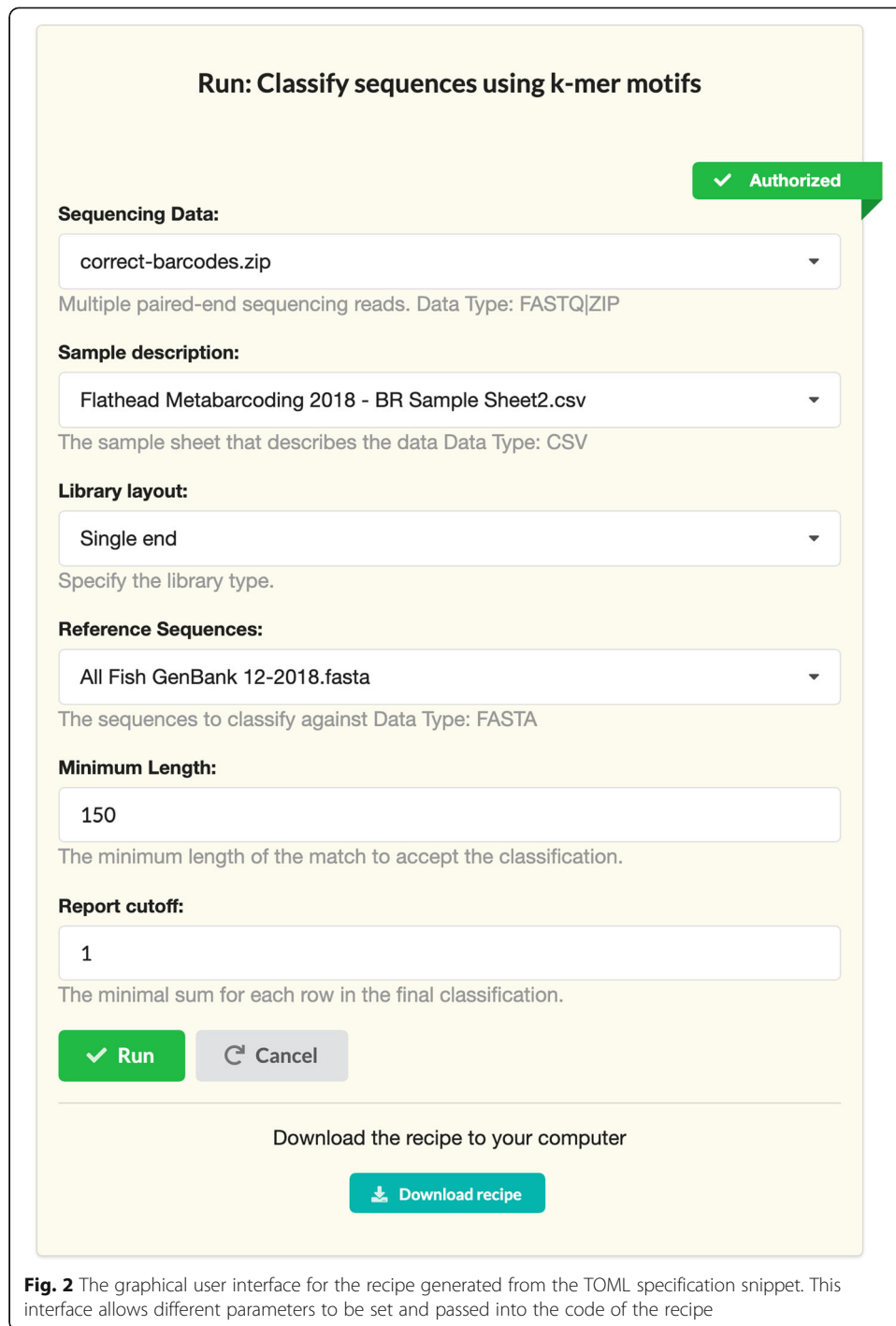


Each recipe may be assigned a graphical user interface specification code in TOML format. From the TOML code the recipe website will generate a user interface, connected to the underlying data analysis script. For example, the TOML code (partially shown) below:

```
[reads]
value = "Fish Metabarcodes"
label = "Sequencing Data"
help = "Multiple paired-end sequencing reads."
source = "PROJECT"
type = "FASTQ |ZIP"
display = "DROPDOWN"

[sheet]
value = "Samplesheet for Fish Metabarcodes"
label = "Sample description"
help = "The sample sheet that describes the data"
source = "PROJECT"
type = "CSV"
display = "DROPDOWN"
```

would generate the interface shown in Fig. 2. When a recipe is executed the parameters selected on the graphical user interface will replace the corresponding parameters inside the recipe. The interface generation “specification language” provides the building blocks for creating user interfaces.



The code for each recipe may be inspected before executing the recipe as seen in Fig. 3. Notably the recipe code consists of executable instructions that may be run on other platforms.

Running a recipe on data entry produces a “result” directory. Result directories consists of all files and all the metadata created by the recipe as it is executed on the input data. Each run of a recipe will generate a *new* result directory. Users may inspect,

Recipe Code

```

15
16 # The reference data in FASTA format.
17 REF=refs/${ACC}.fa
18
19 # Obtain the reference genome.
20 efetch -db nuccore -id $ACC -format fasta > $REF
21
22 # Index the FASTA file for bwa.
23 bwa index $REF 2>> log.txt
24
25 # Index the FASTA file for IGV.
26 samtools faidx $REF >> log.txt
27
28 # Obtain the SRR run number for the data deposited into the BioProject
29 esearch -db sra -query $ID | efetch -format runinfo > runinfo.csv
30
31 # Keep the top N SRR numbers for now.
32 cat runinfo.csv | cut -f 1 -d , | grep SRR | head -N > srr.txt
33
34 # Make a directory for the sequencing data.
35 mkdir -p reads
    
```

Fig. 3 The recipe code consists of the computer code that is executed when the recipe is run. This code may be a series of shell commands, R code, Python code, or any scripting-oriented instruction set

Project 0 Data 21 Recipes 25 Results View Result

MULTI SAMPLE VARIANTS Results for How to generate variant calls for multiple samples: AF086833, PRJNA257197, 5
 This recipe demonstrates variant call generation for several samples at once.
 Completed • Runtime 45 seconds • updated 11 months ago by Istvan Albert

Recipe Rerun Recipe Edit Results Copy Delete

Run Parameters
 Parameters used during the run:

- Genome accession number: AF086833
- The NCBI project run number: PRJNA257197
- How many samples to process: 5

Result description.

File List
 Files created by the recipe run:

runinfo.csv	391.7 KB
srr.txt	55 bytes
variants.vcf	89.1 KB
log.txt	14.5 KB
recipe.sh	1.5 KB
runlog/stdout.txt	0 bytes
runlog/stderr.txt	911 bytes
runlog/input.json	2.2 KB
reads/SRR1972921_1.fastq	26.1 MB
reads/SRR1972918_2.fastq	26.1 MB
reads/SRR1972917_2.fastq	26.1 MB
reads/SRR1972919_2.fastq	26.1 MB

Fig. 4 The result interface shows all the files generated by a recipe run. In addition, the directory contains all the information necessary to reproduce the analysis, the code, the metadata and a log for the standard input and output generated during the execution of the recipe

investigate and download any of the files generated during the recipe run. Additionally, users may copy a result file as new data input for another recipe.

Upon executing a recipe on a dataset, a result directory is generated that lists all files created during the recipe run. See Fig. 4. In addition, all messages printed on the standard output or standard error streams are captured as files and may be inspected later.

The web application that we have developed also provides laboratory data management services. Recipes, data and results can be copied across projects, users may create new projects and may allow others (or the public) to access the contents of a project. As constructed, the web application provides a transparent and consistent framework to conduct analyses that can be shared among collaborators or with the public, and may be reproduced over time due the preservation of runtime-specific version of the code.

Discussion

The need to simplify access to command line tools via graphical user interfaces has been long recognized by several research groups. To address this need various frameworks with similar goals [3–5] have been proposed, developed and deployed. For example, Shiny [6] is an R package that provides a framework for turning R code into an interactive webpage. Webemboss [7] was proposed as a web based environment from which a user can make use of EMBOSS tools in a user-friendly way. Our approach differs substantially from each of these prior works and is aimed to serve the needs of different audiences. Conceptually the most similar software is Galaxy [8], a web application that deploys command-line oriented bioinformatics software tools via a web based graphical user interface.

The recipe approach is similar to Galaxy in that it serves non-technical audiences. Additionally, just like Galaxy, recipes provide a user friendly, graphical interface to facilitate their use. The main difference relative to Galaxy is that, every recipe is downloadable and executable as a standalone program. Thus, recipes may be run on any computational platform and do not depend on the web interface. We refer readers to the detailed documentation of software available at <https://bioinformatics-recipes.readthedocs.io/> where we discuss in detail the differences between our approach and that of existing software platforms.

Notably, in our recipe approach, the roles are more separated and distinct than in Galaxy. In our typical use cases, bioinformaticians develop and test the analysis code at the command line, then they turn their code into recipes and share them with all collaborators. Once shared via the website, collaborators can then select parameters and execute a recipe using data of their choice. Collaborators may inspect, copy, and modify the recipe code.

Conclusion

Our software has been developed to provide bioinformatics support to metabarcoding analyses at the US Fish and Wildlife Northeast Fisheries Center and has been in operation for over a year. We found that the software is well suited for environments where bioinformaticians interact and collaborate with scientists from diverse backgrounds, and when consistent types of analyses need to be applied to varying datasets. In addition, we have found that the recipe approach integrates well into bioinformatics education. We have made use of the bioinformatics recipes website while delivering graduate level classes over several semesters at Penn State and found the approach to be well received by students. Using recipes allowed us to

demonstrate the use of bioinformatics software in a manner that closely resembles their original command line usage. As instructors we were able to demonstrate complete workflows to students, show both the code and all the results that the code produced, while allowing students to copy, share and customize computational pipelines.

The current deployment contains a tutorial, education related materials as well as numerous recipes that demonstrate typical analytical workflows from quality control to RNA-Seq data analysis. We envision individual research groups and organizations running their private instances of our code to serve their local needs and audiences. Using this web platform to host the software allows the various bioinformatics analysis tools as well as the code used as part of the pipeline to be updated as new versions are available.

In conclusion, we have developed a software that supports bioinformaticians assisting and collaborating with life scientists. The software is presented via a web application that provides a high utility and user-friendly tool for conducting reproducible research.

Acknowledgements

Not applicable.

Availability and requirements

Project name: Biostar Central
Project home page: <https://github.com/ialbert/biostar-central>
Documentation: <https://bioinformatics-recipes.readthedocs.io>
Operating system(s): Platform independent
Programming language: Python 3.6 or above
Other requirements: Django
License: GNU GPL 3.0
Restrictions to use by non-academics: none.

Authors' contributions

AM, CR, MB and IA conceived the project. NA and IA developed the software and wrote the paper. AS, AM and IA developed recipes for metabarcoding analysis. All authors contributed to the functional design and usability design of the software. IA supervised the research. All authors read and approved the final manuscript.

Funding

This work has been supported by the US Fish and Wildlife Service Cooperative Agreement Award F16AC01007 as well as the Pennsylvania State University. The funds supported the cost of developing and deploying the software application and the cost of developing data analysis recipes for the Northeast Fisheries Center of the US Fish and Wildlife Service. The findings and conclusions in this article are those of the author(s) and do not necessarily represent the views of the U.S. Fish and Wildlife Service.

Availability of data and materials

A public deployment of the Bioinformatics Recipes software can be accessed at <https://www.bioinformatics.recipes/>. The website contains recipes developed for US Fish and Wildlife as well as recipes used in online courses teaching bioinformatics. The code is released with an open source license and may be accessed at: <https://github.com/ialbert/biostar-central>.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

Not applicable.

Author details

¹Department of Biochemistry and Molecular Biology, Pennsylvania State University, 201 Old Main, University Park, PA 16802, USA. ²Northeast Fisheries Center, US Fish and Wildlife Service, Lamar, PA 16848, USA.

Received: 10 September 2019 Accepted: 12 June 2020
Published online: 08 July 2020

References

1. Strozzi, et al. Scalable workflows and reproducible data analysis for genomics. *Methods Mol Biol.* 2019;1910:723–45.
2. Leipzig J. A review of bioinformatic pipeline frameworks. *Brief Bioinform.* 2017;18(3):530–6. <https://doi.org/10.1093/bib/bbw020>.
3. Federico A, et al. Pipeliner: a Nextflow-based framework for the definition of sequencing data processing pipelines, *Front Genet.* 2019;10:614. <https://doi.org/10.3389/fgene.2019.00614>.
4. Tommaso P. Nextflow enables reproducible computational workflows. *Nat Biotechnol.* 2017;35:316–9.
5. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics.* 2012;28(19):2520–2. <https://doi.org/10.1093/bioinformatics/bts480>.
6. W Chang et al (2019), Shiny: web application framework for R, <https://CRAN.R-project.org/package=shiny>.
7. Sarachu M, et al. wEMBOSS: a web interface for EMBOSS. *Bioinformatics.* 2005;18(4):540–1.
8. Giardine B, et al. Galaxy: a platform for interactive large-scale genome analysis. *Genome Res.* 2005;15(10):1451–5.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

