*Bioinformatics*, 20202, 0–0

doi: 10.1093/bioinformatics/xxxxx

Advance Access Publication Date: 27 March 20202

Manuscript Category

|  |
| --- |
| Webpage Toolbox  Web development for biological analyses  David Hidalgo Gil  Department of Biology, box 118,22100, Lund University, Sweden  \*David Hidalgo Gil  Associate Editor: Eran Elhaik  Received on 24 March 20202; revised on 27 March 2020; accepted on 27 March 2020  Abstract  **Motivation:** This webpage/web app has been designed with the idea of easing the workload generated by some basic but daily tasks of a bioinformaticians such us file transfotmation and nucleotide plotting over sequences/chromosomes.  **Results:** This tool consist of a webpage with a friendly user interface to which the users can upload their files and perform file transformations such as mutate multiline fasta files into oneline fasta files, transform fastq files into fasta files or plot the nucleotide abundance across the input file.  **Availability:** [Git Repository](https://github.com/davhg96/WebApp.git)  **Contact:** da4206hi-s@student.lu.se  **Supplementary information:** Supplementary data are available at the github repository. |

# Introduction

In the actual laboratory bioinformaticians perform most of the heavy lifting in terms of analyzing data, but this job also includes several basic workflows such as file parsing and file transformations as an intermediate step between the raw data and the actual analysis. In some cases, the analysis can be performed by our wet lab colleagues but they still need their data to be transformed to a suitable format. This tool has been developed to suit that need.

# Methods

This tool uses *Python (v.3.6)* as the main programing language to code the main functionality, also different libraries were added such us [*Matplotlib*](https://matplotlib.org/) *(v.3.1.3)*  for the plotting functionality.

For the back-end programing the [*Flask library*](https://flask.palletsprojects.com/en/1.1.x/) *(v.1.1.1)* was used and [*Werkzeug*](https://werkzeug.palletsprojects.com/en/1.0.x/utils/) *(v.1.0.0)* provides with some security tools for the querying system.

All the front end was designed in *HTML5+CSS* with [Bootstrap CDN](https://getbootstrap.com/docs/4.4/getting-started/introduction/) (Content Delivery Network)*(v.4.4)* classes

# Features

At the moment the page has 3 different tools:

* **FastQ to FASTA**. This is a file transformation tool to transform FastQ files into fasta files. The user can select a file from their computer and will get a response with the mutated file.
* **Multiline Fasta to oneline**. This tool transforms multiline fasta files into one line fasta files. The user can select a file from their computer and will get a response with the mutated files.
* **Nucleotide plots**. This tool is designed to plot the nucleotide frequency along the required fasta file. The user has to select an input file. Window step and size parameters and the type of desired plot (GC and All nucleotides are the only ones available for now). The software can detect some non-conventional nucleotides and plot them if they are present in the input file.

The webpage approach has advantages and disadvantages against a standalone program. The main advantage is that this approach is compatible across every platform and a standalone program would require specific platform programming and design. On the other hand, this limits the functionality in different ways. A standalone program allows for a better file management on the user’s local computer so there is no need to rent online storage for the user system and database.

For this reason the page stores the files, generates and answer and deletes everything from the server storage. This might suppose extra strain on the server’s hardware.

## Plot output example

Figure 1. Result plot generated by the web application using “dymmy.fasta” file as an input, window size = 20, windowstep=10. The input file can be found in the [Git Repository](https://github.com/davhg96/WebApp.git)

Discussion

This web application for the moment has some very basic functionality, but it can be escalated and add all sorts of functionality. Also the smart front-end design allows for easy re-scaling across almost every screen size and resolution.

For now the main limitation is that the user can only input one file at a time, and the code can probably be optimized to run faster.

We foresee a good future for this kind of tools that enable everyone to perform bioinformatics tasks without the fear of a command line.

Acknowledgements

We are grateful to our teacher personal who led us to this point in our masters, to Lund university that gave us this opportunity and to our beautiful classmates that attended our presentation in times of corona virus.

References

Alexandrescu,A. (2001) Modern C++ Design: Generic Programming and Design Patterens Applied. Addision Wesley Professional, Boston.