easyfm: An easy software suite for file manipulation of Next Generation Sequencing data on desktops

Hyungtaek Jung[1], Brendan Jeon[1], and Daniel Ortiz-Barrientos[2]

[1]The University of Queensland
[2]The University of Queenland

April 16, 2024

Abstract

Storing and manipulating Next Generation Sequencing (NGS) file formats for understanding biological phenomena is an essential but difficult task in the life sciences. Yet, most methods for analysing NGS data require complex command-line tools in high-performance computing (HPC) or web-based servers and have not yet been implemented in comprehensive, easy-to-use software. Here we present easyfm (easy file manipulation), a free standalone Graphical User Interface (GUI) software with Python support that can be used to facilitate the rapid discovery of target sequences (or user's interest) in NGS datasets for novice users (more accessible to biologists). It enables them to perform end-to-end reproducible data analyses using a desktop application (Windows, Mac and Linux). Unlike existing tools, the GUI-based easyfm is not dependent on any HPC system and can be operated without an internet connection. For user-friendliness and convenience, easyfm was developed with four work modules and a secondary GUI window, covering different aspects of NGS data analysis, including post-processing, filtering, format conversion, generating results, real-time log, and help. In combination with the executable tools (BLAST+ and BLAT) and Python, easyfm allows the user to set analysis parameters, select/extract regions of interest, examine the input and output results, and convert to a wide range of file formats. To help augment the functionality of existing web-based and command-line tools, easyfm, a self-contained program, comes with extensive documentation (https://github.com/TaekAndBrendan/easyfm). This specific benefit allows easyfm to seamlessly integrate visual and interactive representations of NGS files, supporting a wider scope of bioinformatics applications in the life sciences.

1 INTRODUCTION

With the broad implementation of NGS technologies in the life sciences, genomics and transcriptomics sequencing data are generated at an unprecedented rate (Breese & Liu, 2013; Jung et al., 2019; Jung et al., 2020). Rapid progress in NGS technologies has brought massively high-throughput sequencing data to support research questions across many research fields, enabling a new era of genomic research (Jung et al., 2019; Jung et al., 2020). Simultaneously, this advancement has brought enormous challenges in data analysis, of which efficient, standardized and consistent analysis are fundamental steps for maintaining reproducibility, especially for biologists (Breese & Liu, 2013; Jung et al., 2019; Jung et al., 2020). However, many of the available tools for NGS data analysis require higher-order computational experience (e.g. various programming/scripting languages), expensive infrastructure (adequate HPC facilities and Cloud computing) and lack GUIs, making them inaccessible to many researchers, and cumbersome for even experienced biologists. Thus, the development of user-friendly standalone software for NGS data will accelerate the pace of research for scientists who have limited computer and bioinformatics experience.

NGS data processing often involves consecutive steps of trimming (including quality check), assembling, mapping, manipulating, converting and processing large files. FASTA (Pearson & Lipman, 1988) and FASTQ (Cock et al., 2010) file formats are generated by most NGS platforms, and further SAM/BAM (Li et al., 2009), BED (Kent et al, 2002), GFF/GTF (Pertea & Pertea, 2020), and VCF (Danecek et al., 2011) can be derived using FASTA and FASTQ files depending on the required analysis. The FASTA file, based on simple text, is the most basic format for reporting a sequence and is accepted by almost all sequence analysis programs. Each sequence starts with a ">" followed by the sequence name, a description of the sequence, and the sequence itself (nucleic acids or amino acids). The FASTQ file, a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores, is the most widely used format in sequence analysis and NGS sequencers. Each sequence requires at least 4 lines starting with "@" followed by the sequence, a "+" sequence identifier, and quality scores. Conveniently, FASTQ files can also be converted to FASTA files, the most commonly used file format for NGS data that enables direct sequencing of target genes. Many available tools, easySEARCH (Kim et al., 2012); BlasterJS (Blanco-Míguez et al., 2018); BlastGUI (Du et al., 2020); Sequenceserver (Priyam et al., 2019); orfipy (Singh & Wurtele, 2021); Samtools and BCFtools (Danecek et al., 2021) including *easyfm* , have not surprisingly focused on manipulating (analyse, collect, organise, interpret, and present data in meaningful ways) the FASTA file format to generate biologically relevant insights.

For the last decade, many HPC and Cloud-based NGS command-line programs or web-based platforms have wrapped popular high-level analysis and visualisation tools in an intuitive and appealing interface (Baker et al., 2020). Galaxy (homepage: https://galaxyproject.org, main public server: https://usegalaxy.org, Australia: https://usegalaxy.org.au/) in particular has been successful in establishing itself as an analytics hub and an e-learning platform with global scientists, intending to produce accessible, reproducible and collaborative biological analyses (Afgan et al., 2018; Serano-Solano et al., 2021). Even with the huge achievements made in many analytical software packages and pipelines, further improvements in user-friendly standalone software are still required to facilitate the rapid discovery of meaningful sequences in very large data sets for novice users. To help augment the functionality of existing tools and allow for user-friendliness and convenience of NGS file manipulation,*easyfm* enables end-to-end file filtering, extracting and converting (FASTQ to FASTA) with a simple mouse click on desktops.

The *easyfm* , implemented in Python 3.7+, was developed with four work modules (Basic Local Alignment Search Tool [BLAST], BLAST-Like Alignment Tool [BLAT], Open Reading Frames [ORF], and File Manipulation) and a secondary window (Project Folder, Help and Log). Together, these modules and secondary window cover different aspects of NGS data analysis (mainly focusing on FASTA files), including post-processing, filtering, format conversion, and generating results. The functionality of each module has been described in the Results and Discussion section to have an easy-to-follow parallel comparison.*easyfm* is a GUI-based, lightweight but powerful, free and open-source desktop software for querying/manipulating NGS data sources and generating various outcomes. Since everyone can use it from anywhere to analyse data and find target sequences easily without any coding, HPC and/or internet/web-server connection, we hope the usefulness of*easyfm* can extend its potential use in a wide range of bioinformatics applications in the life sciences including teaching/learning materials in the classroom.

2 MATERIALS AND METHODS

easyfm can be used both by sophisticated data scientists and non-technical users who need an intuitive interface. The original intent for producing *easyfm* was to reduce reliance on any command lines/scripts or web-based platforms, by creating a standalone lightweight program with substantially reduced computational demands.*easyfm* provides key benefits in convenience, accessibility, and reproducibility because it does not include any heavyweight NGS data assembly, mapping and clustering workflows. *easyfm* can execute any pre-assembled genome/transcriptome FASTA files by selecting CPU numbers on a user's desktop. While it mainly focuses on point-and-click analysis for less technical users, Log and Help functions could provide an interactive experience for monitoring and iterating on an executed code.

The *easyfm* work modules can provide support for post-processing, filtering, format conversion, and generating results to your given data (e.g. FASTA/Q files). It integrates four Python libraries and two executable programs with additional visualisation and conversation tools (mostly many well-established open-source Python packages) (Table 1). BLAST and indexing features provide the foundation for *easyfm*with approaches for all four work modules (BLAST, BLAT, ORF, and File Manipulation). While the user is required to select a module to execute, the user has full control over which input (including compressed files: *.gz) and output files/folders can be selected. *easyfm* also generates several output files (mostly in a tab-separated text file) that can be opened with standard text editors or Excel. To support work modules, *easyfm* also has a secondary window— Project Folder, Help and Log— that integrates with work modules (Figure 1). In addition, further assistance and information can be obtained via Help and Log to improve processes and performance. *easyfm* also contains all necessary dependencies. Simply unzip the folder and double-click easyfm.exe after downloading the program. Documentation, along with tutorials, is available at https://github.com/TaekAndBrendan/easyfm, and links to the*easyfm* download (https://github.com/TaekAndBrendan/easyfm/raw/main/windows/easyfm.7z).

TABLE 1. Software packages integrated into *easyfm* and their applications

Software	**Application**
Python Packages	Python Packages
Biopython (Cock et al., 2009)	Biopython is a set of freely available tools for common bioinformatics tasks inc
PyQt5 (https://pypi.org/project/PyQt5/)	PyQt is a Python binding of the cross-platform GUI toolkit Qt.
gffutils (https://github.com/daler/gffutils)	gffutils is for working with and manipulating the GFF and GTF format files ty
Pyfastx (Du et al., 2021)	The pyfastx is a lightweight Python C extension that enables users to randoml
Executable Programs	Executable Programs
BLAST+ (v2.11.0) (Camacho et al., 2009)	BLAST+ is a sequence similarity searching tool with an enhancement of speed
BLAT (v3.2.1) (Altschul, 1997)	BLAT is an alignment tool like BLAST and is useful for aligning long sequenc

Please note that *easyfm* , a self-contained program, includes all necessary dependencies and executable packages. Simply unzip the folder and double-click easyfm.exe after downloading the program.

3 RESULTS AND DISCUSSION

3.1 Practical integration of secondary window in *easyfm*

To maximise the capability of the work modules (BLAST, BLAT, ORF, and File Manipulation), *easyfm* provides a secondary window, containing the tabs Project Folder, Help and Log to enhance the intuitive interface and interactive experience. As illustrated in Figure 1, the secondary window GUI components are freely adjustable with mouse movement (four corners) and are seamlessly integrated with the main work module of *easyfm* . The user can control input and output files by selecting the work folder (Project Folder and Set Project in Figure 1B green box) to use work modules. While the user can start with the default folder or select a specific work folder via Project Folder in the local drive, the input files must be in the designated folder. If the files (including compressed files: *.gz) are available in Project Folder, a simple right mouse click on the file can offer more options, such as Get Fasta Information (Stats), Open with Text Editor, Delete, and Create Folder (Figure 1B and 1C). The Help option is a resource intended to provide the end-user with information and support to*easyfm* work modules including its manual. To access additional information the user can click any of the links in Help (Figure 1D). Furthermore, to combine advanced functionalities with an easy-to-use interface, the Log option provides real-time log reporting and monitoring for every executed job (Figure 1D). This can aid in effective communication when reporting and resolving any program issues.

Hosted file

`image1.emf` available at https://authorea.com/users/449177/articles/712530-easyfm-an-easy-software-suite-for-file-manipulation-of-next-generation-sequencing-data-on-desktops

FIGURE 1

Integration of secondary window with main work module of *easyfm* . A) Four main work modules (green box) to BLAST, BLAT, ORF, and File Manipulation. B) Three secondary modules (green box) to assist with main work modules and extra features using a right mouse click. C) Fasta file stats information accessed from B. D) Adjustable secondary window (Help and Log) on the top and bottom.

3.2 Intuitive interface of work modules in *easyfm*

To provide an integrated solution for NGS data file manipulation,*easyfm* provides an open-source tool with an easy installation and setup without relying on any web-based server or commercial licences.*easyfm* also allows users to consolidate the import/export data in FASTA/Q format (e.g. *.gz) under four work modules (BLAST, BLAT, ORF, File Manipulation) with an easy step-by-step process. *easyfm* is distributed under the MIT licence as all-in-one installer packages that contain all necessary software tools plus a manual explaining the analysis workflows step-by-step (https://github.com/TaekAndBrendan/easyfm).

3.2.1 BLAST

BLAST is the most well-known analytics tool in life sciences and has become an essential program in every branch of biology to find regions of local similarity between biological (protein or nucleotide) sequences (Altschul et al., 1997; Cock et al., 2009). While the web-based National Center for Biotechnology Information (NCBI) BLAST suite of programs provides comprehensive sequence comparison, it is a major bottleneck due to delayed new data submission with embargo issues (including user-specific new data) and public availability on central BLAST repositories. Fortunately, BLAST can be installed and run locally, but its usage can be challenging for biologists who have limited experience of command-line interfaces. Furthermore, purchasing commercial software of a rich GUI-standalone tool (e.g. CLC Genomic Workbench and Geneious) and its licences is too expensive for many researchers and laboratories. To resolve these matters, *easyfm* provides a new Python-based free GUI for BLAST and more (Figure 2). Users can explore all BLAST+ (v2.11.0) features by creating a local database from which output format can be selected for including controlling analyses parameters and CPU cores. Even a common BLAST archive format (ASN.1) can be converted to any BLAST output format via Result Convert (Figure 2D). To save storage space and enable faster downstream analysis, a BLAST extensible markup language (XML) format (even generated externally) can be converted into a more compact form (e.g. a human-readable csv file) via XML Parsing (Figure 2E). Along with recent free tools (Blanco-Míguez., et al2018; Kim et al., 2012; Priyam et al., 2019), *easyfm* BLAST enables easy and seamless integration of visual and interactive representations of BLAST outputs supporting sequence similarity search. In particular,*easyfm* offers support for creating/searching a local database, changing format and parsing XML files as a standalone cross-platform application. This comprehensive and autonomous interface makes*easyfm* unique when compared to other free existing tools which need to rely on several different web servers.

Hosted file

`image2.emf` available at https://authorea.com/users/449177/articles/712530-easyfm-an-easy-software-suite-for-file-manipulation-of-next-generation-sequencing-data-on-desktops

FIGURE 2

User-friendly standalone work modules in *easyfm* : BLAST module. Most steps include further manual options for a user-specified parameter. A) Create a local database by selecting nucleotide or protein. B) Job completion message and created database files listed in a secondary window. C) Run local BLAST with multiple features including output type. D) Convert from a BLAST archive file to a different output format. E) A BLAST xml file parsing with multiple options for a csv file.

3.2.2 BLAT

BLAT is one of the alignment algorithms developed for the pairwise analysis and comparison of biological sequences with the primary goal of inferring homology to discover the biological function of genomic sequences (Kent, 2002). While BLAT is less sensitive than BLAST, BLAT has a few clear advantages over BLAST from a practical standpoint in speed and convenience (Bhagwat et al., 2012). Compared to pre-existing pairwise sequence alignment tools, BLAT performed ~500 times faster with mRNA/DNA alignments and

~50 times faster with protein/protein alignments (Kent, 2002). BLAT can be used either as a web-based server-client program (https://genome.ucsc.edu/cgi-bin/hgBlat) or as a standalone command-line program (Bhagwat et al., 2012), but not a user-friendly GUI. However, *easyfm* BLAT (v3.2.1) enables users to control all parameters with a simple mouse click (Figure 3A) that can be a great advantage for novice biologists. Along with freely available*easyfm* BLAST, *easyfm* BLAT will simplify distributed computation pipelines to facilitate the rapid discovery of sequence similarities between NGS datasets. However, if the target genome and input sequences are big, using the standalone command-line BLAT in HPC is more suitable for batch runs, and more efficient than the web- and GUI-based BLAT because the standalone command-line in HPC can store more memory.

Hosted file

`image3.emf` available at https://authorea.com/users/449177/articles/712530-easyfm-an-easy-software-suite-for-file-manipulation-of-next-generation-sequencing-data-on-desktops

FIGIRE 3

User-friendly standalone work modules in *easyfm* : BLAT module. Most steps include further manual options for a user-specified parameter. Create and run a local database with multiple options for a psl file that can open with text editor and Excel.

3.2.3 ORF

An ORF(s) is the part of a reading frame that can be translated. The ORF (potential protein-coding sequence) is a continuous stretch of codons that usually begins with a start codon and ends at a stop codon. Understanding ORF(s) has become a piece of essential evidence to assist in gene prediction. As with other ORF finding tools, *easyfm*performs a six-frame translation of a nucleotide given a particular genetic code, finding all ORFs possible. Long ORFs are often used, along with other evidence, to initially identify candidate protein-coding regions or functional RNA-coding regions in a given DNA sequence, but the presence of an ORF does not necessarily mean that the region is always translated (Deonier et al., 2005). As BLAST and BLAT, the web-based ORF Finder (https://www.ncbi.nlm.nih.gov/orffinder/), ORF Predictor (http://bioinformatics.ysu.edu/tools/OrfPredictor.html) and command-line tools (ORF Investigator (Dwivedi & Mishra, 2012) and orfipy (Singh & Wurtele, 2021) offer a range of ORF searches, but its usage can be challenging for biologists due to lack of computer programming literacy and limited query sequence length. To maximise the flexibility, the *easyfm* ORF provides a fast and efficient approach for all possible translation and extraction of ORFs from nucleotide sequences (FASTA format of nucleotide and protein output from six-frame translation) (Figure 4). With a simple mouse click solution, users can compare the translated outcomes with their biological evidence to avoid false discovery as well as control specific parameters without any limitation of query sequence length. Along with existing tools (Dwivedi et al., 2012; Singh & Wurtele, 2021), *easyfm* ORF will provide rapid, flexible searches in multiple output formats to allow the easy downstream analysis of ORFs.

Hosted file

`image4.emf` available at https://authorea.com/users/449177/articles/712530-easyfm-an-easy-software-suite-for-file-manipulation-of-next-generation-sequencing-data-on-desktops

FIGURE 4

User-friendly standalone work modules in *easyfm* : ORF module. Most steps include further options for a user-specified parameter. Run ORF with different genetic codes for coding and protein sequences. A FASTA format output file of nucleotide and protein from a six-frame translation will be generated.

3.2.4 File Manipulation

Various file formats have been introduced with the development of different DNA/RNA sequencing technologies. While there are many different biological file formats related to NGS analyses (or to store and manipulate), FASTA/Q files are most commonly encountered in the bioinformatics community. This is due to their flexibility: FASTA/Q files can be read, mapped and indexed by several different software packages to generate SAM/BAM, GFF/GTF, VCF, and more. Using a fai index file in conjunction with a FASTA/Q file containing reference sequences enables efficient access to arbitrary regions within those reference sequences and extracts subsequences from the indexed reference sequence (Danecek et al. 2021; Quinlan & Hall 2010).

Like other modules, the web-based Galaxy (homepage: https://galaxyproject.org, main public server: https://usegalaxy.org, Australia: https://usegalaxy.org.au/) and command-line tools, Samtools and BCFtools (Danecek et al., 2021) and BEDTools (Quinlan & Hall, 2010), offer a range of NGS data file manipulation capabilities, but its usage can be challenging for biologists due to lack of computer language literacy and internet dependence. To enhance and extend the flexibility and convenience, we present *easyfm* , a free single GUI for NGS file manipulation (mainly for FASTA files) (Figure 5). Since users can control everything with a simple mouse click on a desktop, the tools available in the*easyfm* would be a convenient way to teach bioinformatics/data analysis, and to quickly analyse results without being hampered by command line tools and HPC Secure Shell (SSH) connections.

Users can import any FASTA/Q files to index and extract the indexed ID with its sequence by double-clicking, matching Prefix ID and selecting a provided text file (Figure 5A). Even the FASTQ file can be converted to the FASTA file and the given FASTA file change its direction via Reverse Complement and Reverse (Figure 5B and 5C). For wide applications,*easyfm* File Manipulation also allows users to easily manipulate (including filtering [IDs, features and strand] and extracting sequence regions) and consolidate from GFF and GTF files if its corresponding reference genome/transcriptome sequences are present (Figure 5D). To enhance user-friendliness, users can extract a given sequence as a FASTA file with extra flanking regions for both directions by entering the desired sequence length (numeric numbers). Along with existing tools (Danecek et al., 2021; Quinlan & Hall, 2010),*easyfm* File Manipulation will provide a stable and modular platform for manipulating sequence data and files to ensure high reproducibility standards in the NGS era.

Hosted file

`image5.emf` available at https://authorea.com/users/449177/articles/712530-easyfm-an-easy-software-suite-for-file-manipulation-of-next-generation-sequencing-data-on-desktops

FIGIRE 5

User-friendly standalone work modules in *easyfm* : File Manipulation module. Most steps include further individual selection by manually saving as a FASTA file for a user-specified sequence ID. A) Select a FASTA file to index. B) Convert nucleotide sequences for reverse complement or just reverse sequence. C) Convert and extract from FASTQ to FASTA. D) Extract sequences with specific IDs from indexed reference FASTA and GFF3/GTF files with different features.

FUTURE IMPLEMENTATIONS

easyfm is implemented in Python and available under the MIT license and works on Windows, Linux and Mac systems. This package is also available on PyPI python package manager. The current code runs under Python 3.7+ and virtualenv. Other dependency includes gffutils, pyfastx, PyQt5 and Biopython (Table 1). More information and the manual may be obtained from the website: https://github.com/TaekAndBrendan/easyfm.

In the future, we will continue to update the toolbox with new fast and easy GUI support, including new embedding methods such as DIAMOND (Buchfink et al., 2015; Buchfink et al., 2021) and pBLAT (Wang & Kong, 2019) with low resource requirements and both multithread and cluster computing support, making these methods suitable for running on standard desktops and laptops. Future versions of *easyfm* will also include additional integration points allowing us to intersect, merge, count, complement, and shuffle genomic intervals from multiple files in widely-used genomic file formats such as BAM, BED and VCF.

ACKNOWLEDGEMENTS

The authors are grateful to Maddie James, Melanie Wilkinson, Cara Conradsen, and Nicholas O'Brien for their valuable and constructive comments. This work was supported by ARC Discovery (DP190103039) to D.O.

AUTHOR CONTRIBUTIONS

H.J. and B.J. conceived and designed the study and wrote the Python package. H.J., B.J. and D.O. wrote the paper. All authors read and approved the final manuscript.

DATA AVAILABILITY STATEMENT

The Python package 3.7+ *easyfm* and its manual are available from GitHub: https://github.com/TaekAndBrendan/easyfm. The latest version is available from https://github.com/TaekAndBrendan/easyfm/raw/main/windows/easyfm.7z. The scripts and the data used in the examples or in the demonstration studies of this manuscript are available from a GitHub repository https://github.com/TaekAndBrendan/easyfm/blob/main/easyfm.py and https://github.com/TaekAndBrendan/easyfm under MIT License. This study is not required to deposit any raw sequence reads, individual genotype data and unique haplotype data on SRA/DataDryad/NCBI Nucleotide Database.

References

Afgan, E., Baker, D., Batut. B., van den Beek, M., Bouvier, D., Čech, M., & et al (2018). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research* , **46** (W1), W537–W544.

Altschul, S. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* ,**25** (17), 3389–3402.

Baker, Q. B., Hammad, M. M., Al-Rashdan, W., Jararweh, Y., Al-Smadi, M., & Al-Zinati, M. (2020). Comprehensive comparison of Cloud-based NGS data analysis and alignment tools. *Informatics in Medicine Unlocked* , **18** , 100296.

Bhagwat, M., Young, L., & Robison, R. (2012). Using BLAT to find sequence similarity in closely related genomes. *Curr Protoc Bioinformatics* , **Chapter 10** , Unit 10.8.

Blanco-Míguez, A., Fdez-Riverola, F., Sánchez, B., & Lourenço, A. (2018). BlasterJS: A novel interactive JavaScript visualisation component for BLAST alignment results. *PLoS One* , **13** (10), e0205286.

Breese, M. R., & Liu, Y. (2013). NGSUtils: a software suite for analyzing and manipulating next-generation sequencing datasets.*Bioinformatics* , **29** (4), 494–496.

Buchfink, B., Xie, C., & Huson, DH. (2015). Fast and sensitive protein alignment using DIAMOND. *Nature Methods* , **12** (1), 59–60.

Buchfink, B., Reuter, K., & Drost, H.G. (2021). Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nature Methods* ,**18** (4), 366–368.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & et al. (2009). BLAST+: architecture and applications.*BMC Bioinformatics* , **10** , 421.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., & et al. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics.*Bioinformatics* , **25** (11), 1422–1423.

Cock, P. J. A., Fields, C. J, Goto, N., Heuer, M. L., & Rice, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research* ,**38** (6), 1767–1771.

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., & et al. (2011). The variant call format and VCFtools.*Bioinformatics* , **27** (15), 2156–2158.

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., & et al. (2021). Twelve years of SAMtools and BCFtools.*Gigascience* , **10** (2), giab008.

Deonier, R., Tavaré, S., & Waterman, M. (2005). Computational Genome Analysis: an introduction. *Springer-Verlag* , **25** .

Du. Z., Wu, Q., Wang, T., Chen, D., Huang, X., Yang, W., & Luo, W. (2019). BlastGUI: A python-based cross-platform local BLAST visualization software. *Molecular Informatics* , **39** (4), 1900120.

Du, L., Liu, Q., Fan, Z., Tang, J., Zhang, X., Price, M., & et al. (2021). Pyfastx: a robust Python package for fast random access to sequences from plain and gzipped FASTA/Q files. *Briefings in Bioinformatics* , **22** (4), bbaa368.

Dwivedi, V. D., & Mishra, S.K. (2012). ORF Investigator: A New ORF finding tool combining Pairwise Global Gene Alignment. *Research Journal of Recent Sciences* , **1** (11), 32–35.

Jung, H., Winefield, C., Bombarely, A., Prentis, P., & Waterhouse, P. (2019). Tools and strategies for long-read sequencing and de novo assembly of plant genomes. *Trends in Plant Science* ,**24** (8), 700-724.

Jung, H., Ventura, T., Chung, J., Kim, W.J., Nam, B.H., Kong, H., & et al. (2012). Twelve quick steps for genome assembly and annotation in the classroom. *PLoS Computational Biology* , **16** (11), e1008325.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., & et al. (2002). The human genome browser at UCSC.*Genome Research* , **12** (6), 996–1006.

Kent, W. J. (2002). BLAT–the BLAST-like alignment tool. *Genome Research* , **12** (4), 656–664.

Kim, D. W., Kim, R. N., Kim, D. S., Choi, S. H., Chae, S. H., & Park, H. S. (2012). easySEARCH: A user-friendly bioinformatics program that enables BLAST searching with a massive number of query sequences.*Bioinformation* , **8** (16), 792–794.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., & et al. (2009). The sequence alignment/map format and SAMtools.*Bioinformatics* , **25** (16), 2078–2079.

Pearson, W. R., & Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America* , **85** (8), 2444–2448.

Pertea, G., & Pertea, M. (2020). GFF Utilities: GffRead and GffCompare.*F1000Research* , **9** , 304.

Priyam, A., Woodcroft, B. J., Rai, V., Moghul, I., Munagala, A., Ter, F., & et al. (2019). Sequenceserver: A Modern Graphical User Interface for Custom BLAST Databases. *Molecular Biology and Evolution* ,**36** (12), 2922–2924.

Quinlan, A. R., & Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* ,**26** (6), 841–842.

Singh, U., & Wurtele, E. S. (2021). orfipy: a fast and flexible tool for extracting ORFs. *Bioinformatics* , **37** (28), 3019–3020.

Serrano-Solano, B., Föll, M. C., Gallardo-Alba, C., Erxleben, A., Rasche, H., Hiltemann, S., & et al. (2021). Fostering accessible online education using Galaxy as an e-learning platform. *PLoS Computational Biology* , **17** (5), e1008923.

Wang, M., & Kong, L. (2019). pblat: a multithread blat algorithm speeding up aligning sequences to genomes. *BMC Bioinformatics* ,**20** , 28.