

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

3

4

5

6 **Hyungtaek Jung^{1a*}, Brendan Jeon^{2a*}, Daniel Ortiz-Barrientos¹**

7 ¹ School of Biological Sciences, The University of Queensland, St Lucia, Australia

8 ² Department of Communication and Arts, The University of Queensland, St Lucia, Australia

9

10

11

12 ^a Equal contributions

13 * hyungtaek.jung@uq.edu.au (HJ) and b.jeon@uq.edu.au (BJ)

1 **Abstract**

2 Storing and manipulating Next Generation Sequencing (NGS) file formats is an essential but
3 difficult task in biological data analysis. The *easyfm* (**easy file manipulation**) toolkit
4 (<https://github.com/TaekAndBrendan/easyfm>) makes manipulating commonly used NGS files
5 more accessible to biologists. It enables them to perform end-to-end reproducible data analyses
6 using a free standalone desktop application (available on Windows, Mac and Linux). Unlike
7 existing tools (e.g. Galaxy), the Graphical User Interface (GUI)-based *easyfm* is not dependent
8 on any high-performance computing (HPC) system and can be operated without an internet
9 connection. This specific benefit allow *easyfm* to seamlessly integrate visual and interactive
10 representations of NGS files, supporting a wider scope of bioinformatics applications in the
11 life sciences.

12

13

14 **Author summary**

15 The analysis and manipulation of NGS data for understanding biological phenomena is an
16 increasingly important aspect in the life sciences. Yet, most methods for analysing, storing and
17 manipulating NGS data require complex command-line tools in HPC or web-based servers and
18 have not yet been implemented in comprehensive, easy-to-use software. This is a major hurdle
19 preventing more general application in the field of NGS data analysis and file manipulation.
20 Here we present *easyfm*, a free standalone Graphical User Interface (GUI) software with
21 Python support that can be used to facilitate the rapid discovery of target sequences (or user's
22 interest) in NGS datasets for novice users. For user-friendliness and convenience, *easyfm* was
23 developed with four work modules and a secondary GUI window (herein secondary window),
24 covering different aspects of NGS data analysis (mainly focusing on FASTA files), including
25 post-processing, filtering, format conversion, generating results, real-time log, and help. In
26 combination with the executable tools (BLAST+ and BLAT) and Python, *easyfm* allows the
27 user to set analysis parameters, select/extract regions of interest, examine the input and output
28 results, and convert to a wide range of file formats. To help augment the functionality of
29 existing web-based and command-line tools, *easyfm*, a self-contained program, comes with
30 extensive documentation (hosted at <https://github.com/TaekAndBrendan/easyfm>) including a
31 comprehensive step-by-step guide.

32

1 **1 Introduction**

2 With the broad implementation of NGS technologies in the life sciences, genomics and
3 transcriptomics sequencing data are generated at an unprecedented rate [1–3]. Rapid progress
4 in NGS technologies has brought massively high-throughput sequencing data to support
5 research questions across many research fields, enabling a new era of genomic research [2,3].
6 Simultaneously, this advancement has brought enormous challenges in data analysis, of which
7 efficient, standardized and consistent analysis are fundamental steps for maintaining
8 reproducibility, especially for biologists [1,3]. However, many of the available tools for NGS
9 data analysis require higher-order computational experience (e.g. various
10 programming/scripting languages), expensive infrastructure (adequate HPC facilities and
11 Cloud computing) and lack GUIs, making them inaccessible to many researchers, and
12 cumbersome for even experienced biologists. Thus, the development of user-friendly
13 standalone software for NGS data will accelerate the pace of research for scientists who have
14 limited computer and bioinformatics experience.

15 NGS data processing often involves consecutive steps of trimming (including quality
16 check), assembling, mapping, manipulating, converting and processing large files. FASTA [4]
17 and FASTQ [5] file formats are generated by most NGS platforms, and further SAM/BAM [6],
18 BED [7], GFF/GTF [8], and VCF [9] can be derived using FASTA and FASTQ files depending
19 on the required analysis. The FASTA file, based on simple text, is the most basic format for
20 reporting a sequence and is accepted by almost all sequence analysis programs. Each sequence
21 starts with a “>” followed by the sequence name, a description of the sequence, and the
22 sequence itself (nucleic acids or amino acids). The FASTQ file, a text-based format for storing
23 both a biological sequence (usually nucleotide sequence) and its corresponding quality scores,
24 is the most widely used format in sequence analysis and NGS sequencers. Each sequence
25 requires at least 4 lines starting with “@” followed by the sequence, a “+” sequence identifier,
26 and quality scores. Conveniently, FASTQ files can also be converted to FASTA files, the most
27 commonly used file format for NGS data that enables direct sequencing of target genes. Many
28 available tools (easySEARCH [10]; BlasterJS [11]; Sequenceserver [12]; orfipy [13]);
29 Samtools and BCFtools [14] including *easyfm*) have not surprisingly focused on manipulating
30 (analyse, collect, organise, interpret, and present data in meaningful ways) the FASTA file
31 format to generate biologically relevant insights.

32 For the last decade, many HPC and Cloud-based NGS command-line programs or web-
33 based platforms have wrapped popular high-level analysis and visualisation tools in an intuitive

1 and appealing interface [15]. Galaxy (homepage: <https://galaxyproject.org>, main public server:
2 <https://usegalaxy.org>, Australia: <https://usegalaxy.org.au/>) in particular has been successful in
3 establishing itself as an analytics hub and an e-learning platform with global scientists,
4 intending to produce accessible, reproducible and collaborative biological analyses [16,17].
5 Even with the huge achievements made in many analytical software packages and pipelines,
6 further improvements in user-friendly standalone software are still required to facilitate the
7 rapid discovery of meaningful sequences in very large data sets for novice users. To help
8 augment the functionality of existing tools and allow for user-friendliness and convenience of
9 NGS file manipulation, *easyfm* enables end-to-end file filtering, extracting and converting
10 (FASTQ to FASTA) with a simple mouse click on desktops.

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

24

1 **2 Design and implementation**

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

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

27

28

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

Software	Application
Python Packages	
Biopython [18]	Biopython is a set of freely available tools for common bioinformatics tasks including biological computation.
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 typically used for genomic annotations.
Pyfastx [19]	The pyfastx is a lightweight Python C extension that enables users to randomly access sequences from plain and gzipped FASTA/Q files.
Executable Programs	
BLAST+ (v2.11.0) [20]	BLAST+ is a sequence similarity searching tool with an enhancement of speed and query length.
BLAT (v3.2.1) [21]	BLAT is an alignment tool like BLAST and is useful for aligning long sequences and gapped mapping.

2

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

6

7

8

1 **3 Results and Discussion**

2 **3.1 Practical integration of secondary window in *easyfm***

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

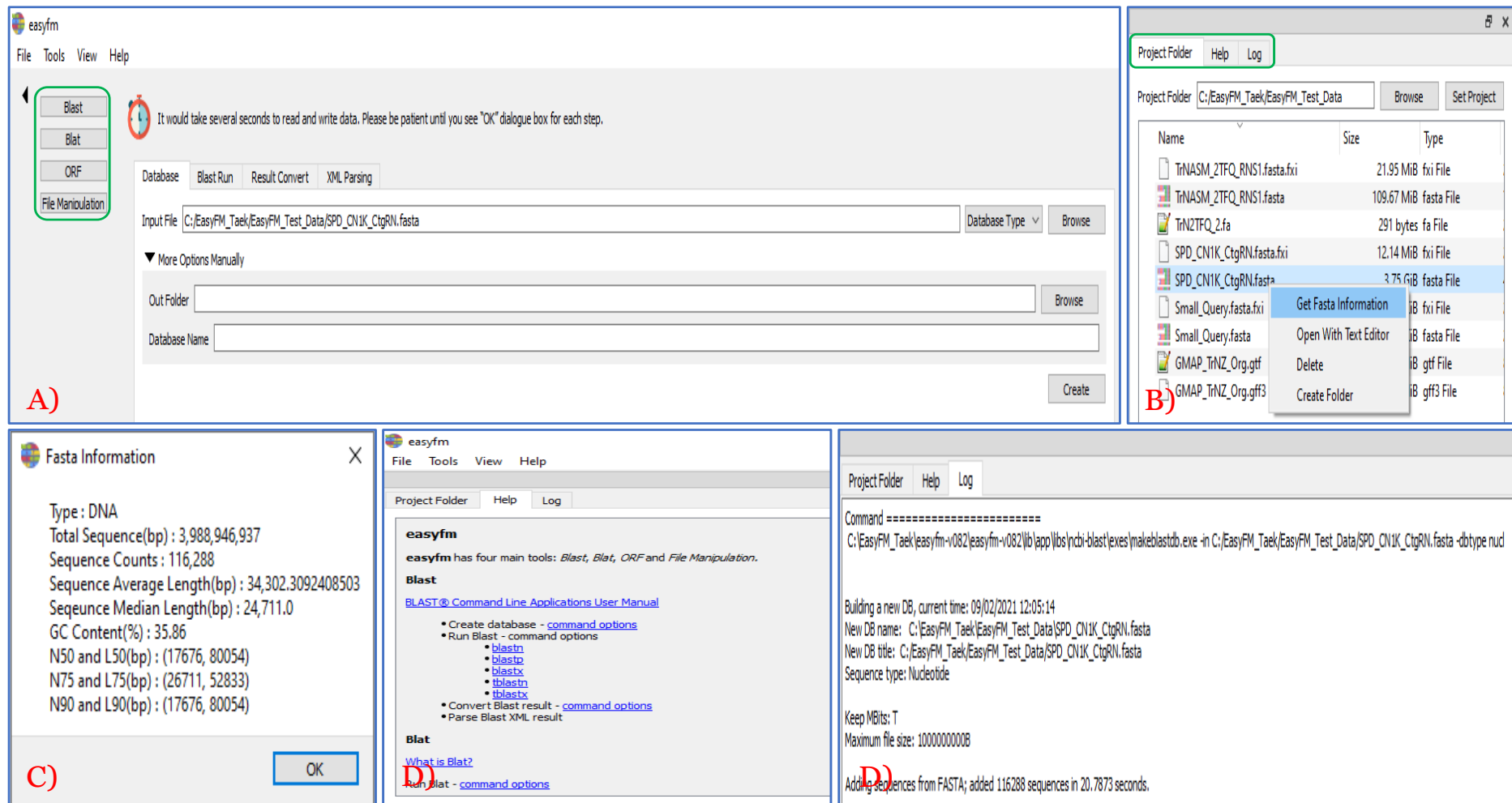


Fig 1. Integration of secondary window with main work module of *easyfm*. A) Four main work modules (green box) to BLAST, BLAT, ORF, and File Manipulation. B) Three secondary modules (green box) to assist with main work modules and extra features using a right mouse click. C) Fasta file stats information accessed from B. D) Adjustable secondary window (Help and Log) on the top and bottom.

1

2 **3.2 Intuitive interface of work modules in *easyfm***

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

10

11 **3.2.1 BLAST**

12 BLAST is the most well-known analytics tool in life sciences and has become an essential
13 program in every branch of biology to find regions of local similarity between biological
14 (protein or nucleotide) sequences [18,22]. While the web-based National Center for
15 Biotechnology Information (NCBI) BLAST suite of programs provides comprehensive
16 sequence comparison, it is a major bottleneck due to delayed new data submission with
17 embargo issues (including user-specific new data) and public availability on central BLAST
18 repositories. Fortunately, BLAST can be installed and run locally, but its usage can be
19 challenging for biologists who have limited experience of command-line interfaces.
20 Furthermore, purchasing commercial software of a rich GUI-standalone tool (e.g. CLC
21 Genomic Workbench and Geneious) and its licences is too expensive for many researchers and
22 laboratories. To resolve these matters, *easyfm* provides a new Python-based free GUI for
23 BLAST and more (Fig 2). Users can explore all BLAST+ (v2.11.0) features by creating a local
24 database from which output format can be selected for including controlling analyses
25 parameters and CPU cores. Even a common BLAST archive format (ASN.1) can be converted
26 to any BLAST output format via Result Convert (Fig 2D). To save storage space and enable
27 faster downstream analysis, a BLAST extensible markup language (XML) format (even
28 generated externally) can be converted into a more compact form (e.g. a human-readable csv
29 file) via XML Parsing (Fig 2E). Along with recent free tools [10–12], *easyfm* BLAST enables
30 easy and seamless integration of visual and interactive representations of BLAST outputs
31 supporting sequence similarity search. In particular, *easyfm* offers support for
32 creating/searching a local database, changing format and parsing XML files as a standalone

- 1 cross-platform application. This comprehensive and autonomous interface makes *easyfm*
- 2 unique when compared to other free existing tools which need to rely on several different web
- 3 servers.

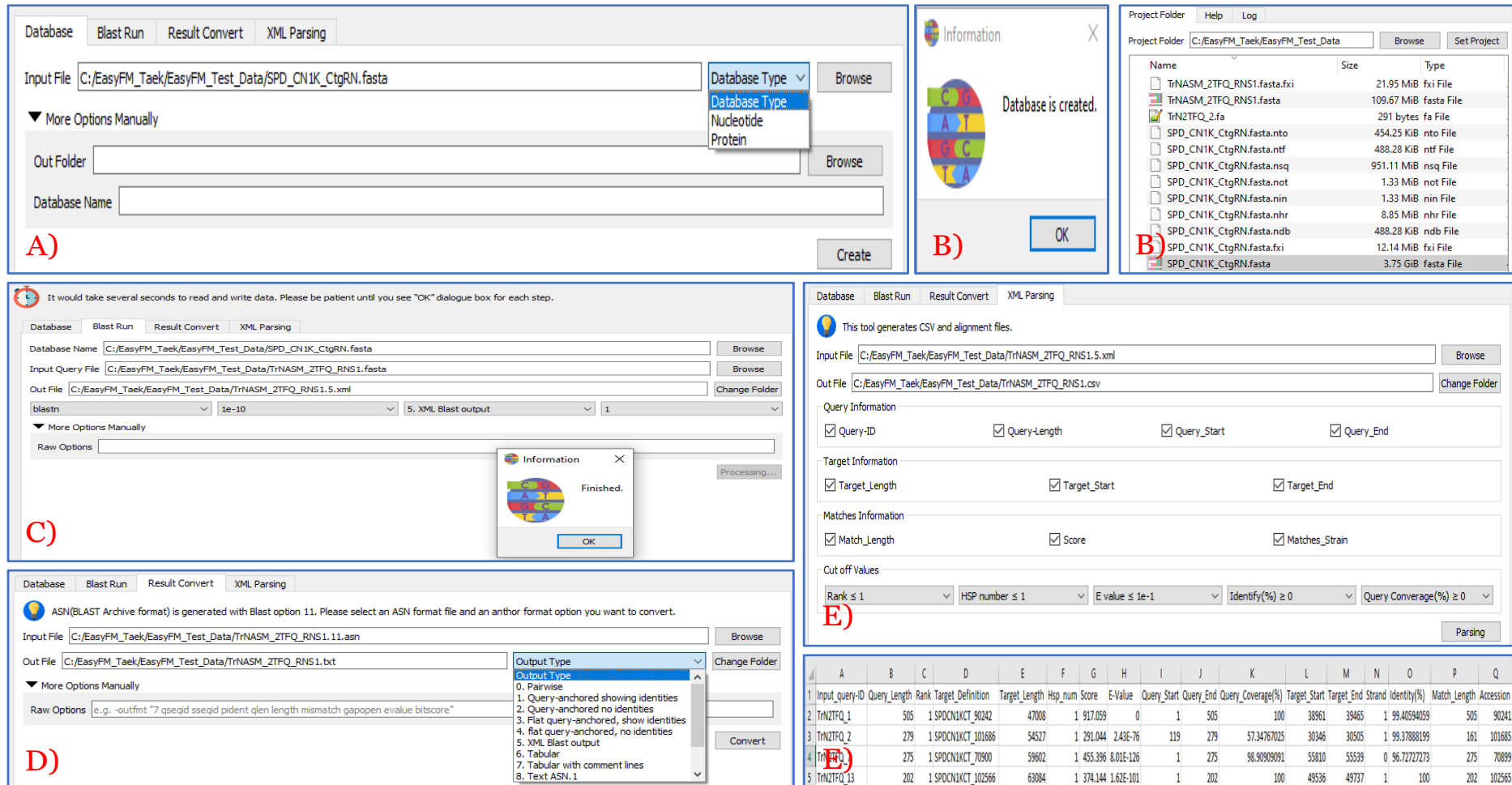


Fig 2. User-friendly standalone work modules in *easyfm*: BLAST module. Most steps include further manual options for a user-specified parameter. A) Create a local database by selecting nucleotide or protein. B) Job completion message and created database files listed in a secondary window. C) Run local 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.

1

2 **3.2.2 BLAT**

3 BLAT is one of the alignment algorithms developed for the pairwise analysis and comparison
4 of biological sequences with the primary goal of inferring homology to discover the biological
5 function of genomic sequences [21]. While BLAT is less sensitive than BLAST, BLAT has a
6 few clear advantages over BLAST from a practical standpoint in speed and convenience [23].
7 Compared to pre-existing pairwise sequence alignment tools, BLAT performed ~500 times
8 faster with mRNA/DNA alignments and ~50 times faster with protein/protein alignments [21].
9 BLAT can be used either as a web-based server-client program ([https://genome.ucsc.edu/cgi-](https://genome.ucsc.edu/cgi-bin/hgBlat)
10 [bin/hgBlat](https://genome.ucsc.edu/cgi-bin/hgBlat)) or as a standalone command-line program [23], but not a user-friendly GUI.
11 However, *easyfm* BLAT (v3.2.1) enables users to control all parameters with a simple mouse
12 click (Fig 3A) that can be a great advantage for novice biologists. Along with freely available
13 *easyfm* BLAST, *easyfm* BLAT will simplify distributed computation pipelines to facilitate the
14 rapid discovery of sequence similarities between NGS datasets. However, if the target genome
15 and input sequences are big, using the standalone command-line BLAT in HPC is more suitable
16 for batch runs, and more efficient than the web- and GUI-based BLAT because the standalone
17 command-line in HPC can store more memory.

Blat Run

Light weight only (Recommended to use a small set data. For big genome in Database Name, please use each chromosome file, respectively).

Database Name

Input File

Out File

Database Type Query Type Tile Size Step Size One Off

More Options Manually

Options

Fig 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.

1 3.2.3 ORF

2 An ORF(s) is the part of a reading frame that can be translated. The ORF (potential protein-
3 coding sequence) is a continuous stretch of codons that usually begins with a start codon and
4 ends at a stop codon. Understanding ORF(s) has become a piece of essential evidence to assist
5 in gene prediction. As with other ORF finding tools, *easyfm* performs a six-frame translation
6 of a nucleotide given a particular genetic code, finding all ORFs possible. Long ORFs are often
7 used, along with other evidence, to initially identify candidate protein-coding regions or
8 functional RNA-coding regions in a given DNA sequence, but the presence of an ORF does
9 not necessarily mean that the region is always translated [24]. As BLAST and BLAT, the web-
10 based ORF Finder (<https://www.ncbi.nlm.nih.gov/orffinder/>), ORF Predictor
11 (<http://bioinformatics.yzu.edu/tools/OrfPredictor.html>) and command-line tools (ORF
12 Investigator [25] and orfipy [13]) offer a range of ORF searches, but its usage can be
13 challenging for biologists due to lack of computer programming literacy and limited query
14 sequence length. To maximise the flexibility, the *easyfm* ORF provides a fast and efficient
15 approach for all possible translation and extraction of ORFs from nucleotide sequences
16 (FASTA format of nucleotide and protein output from six-frame translation) (Fig 4). With a
17 simple mouse click solution, users can compare the translated outcomes with their biological
18 evidence to avoid false discovery as well as control specific parameters without any limitation
19 of query sequence length. Along with existing tools [13,25], *easyfm* ORF will provide rapid,
20 flexible searches in multiple output formats to allow the easy downstream analysis of ORFs.

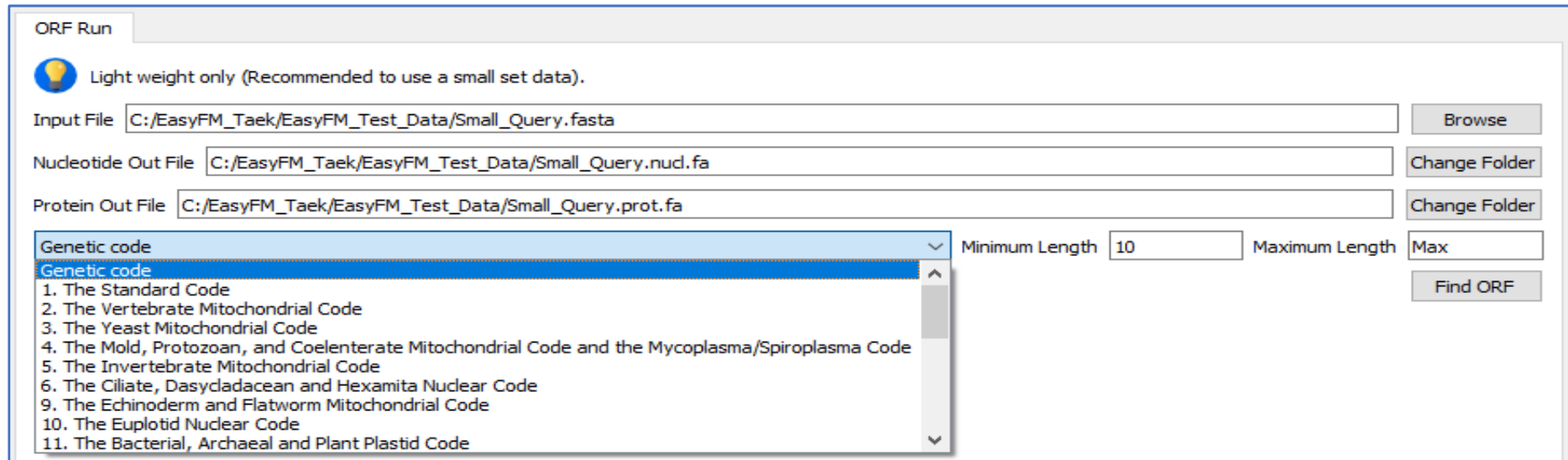


Fig 4. User-friendly standalone work modules in *easyfm*: ORF module. Most steps include further options for a user-specified parameter. Run ORF with different genetic codes for coding and protein sequences. A FASTA format output file of nucleotide and protein from a six-frame translation will be generated.

1 3.2.4 File Manipulation

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

11 Like other modules, the web-based Galaxy (homepage: <https://galaxyproject.org>, main
12 public server: <https://usegalaxy.org>, Australia: <https://usegalaxy.org.au/>) and command-line
13 tools (Samtools and BCFtools [14]; BEDTools [26]) offer a range of NGS data file
14 manipulation capabilities, but its usage can be challenging for biologists due to lack of
15 computer language literacy and internet dependence. To enhance and extend the flexibility and
16 convenience, we present *easyfm*, a free single GUI for NGS file manipulation (mainly for
17 FASTA files) (Fig 5). Since users can control everything with a simple mouse click on a
18 desktop, the tools available in the *easyfm* would be a convenient way to teach
19 bioinformatics/data analysis, and to quickly analyse results without being hampered by
20 command line tools and HPC Secure Shell (SSH) connections.

21 Users can import any FASTA/Q files to index and extract the indexed ID with its
22 sequence by double-clicking, matching Prefix ID and selecting a provided text file (Fig 5A).
23 Even the FASTQ file can be converted to the FASTA file and the given FASTA file change its
24 direction via Reverse Complement and Reverse (Fig 5B and 5C). For wide applications, *easyfm*
25 File Manipulation also allows users to easily manipulate (including filtering [IDs, features and
26 strand] and extracting sequence regions) and consolidate from GFF and GTF files if its
27 corresponding reference genome/transcriptome sequences are present (Fig 5D). To enhance
28 user-friendliness, users can extract a given sequence as a FASTA file with extra flanking
29 regions for both directions by entering the desired sequence length (numeric numbers). Along
30 with existing tools [14,26], *easyfm* File Manipulation will provide a stable and modular
31 platform for manipulating sequence data and files to ensure high reproducibility standards in
32 the NGS era.

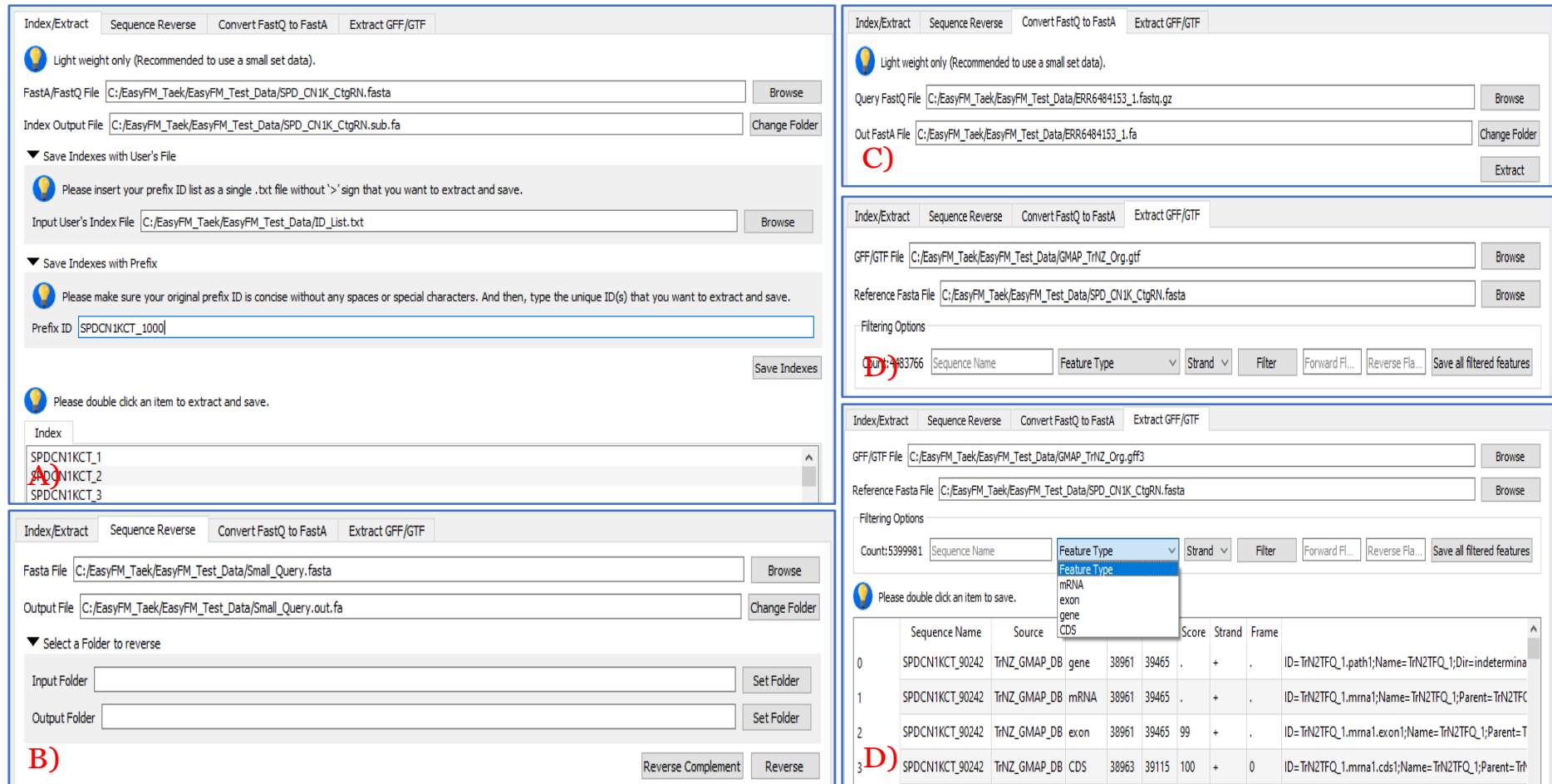


Fig 5. User-friendly standalone work modules in *easyfm*: File Manipulation module. Most steps include further individual selection by manually saving as a FASTA file for a user-specified sequence ID. A) Select a FASTA file to index. B) Convert nucleotide sequences for reverse complement or just reverse 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.

1

2 **Availability and future directions**

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

8 In the future, we will continue to update the toolbox with new fast and easy GUI support,
9 including new embedding methods such as DIAMOND [27,28], (Buchfink et al. 2015, 2021)
10 and pBLAT [29] with low resource requirements and both multithread and cluster computing
11 support, making these methods suitable for running on standard desktops and laptops. Future
12 versions of *easyfm* will also include additional integration points allowing us to intersect, merge,
13 count, complement, and shuffle genomic intervals from multiple files in widely-used genomic
14 file formats such as BAM, BED and VCF.

1 **Acknowledgements**

2 The authors are grateful to their colleagues (special thanks to Maddie James, Melanie
3 Wilkinson, Cara Conradsen, and Nicholas O'Brien) and collaborators for their valuable and
4 constructive comments.

5

1 **References**

- 2 1. Breese MR, Liu Y. NGSUtils: a software suite for analyzing and manipulating next-
3 generation sequencing datasets. *Bioinformatics*. 2013;29:494–496.
- 4 2. Jung H, Winefield C, Bombarely A, Prentis P, Waterhouse, P. Tools and strategies for long-
5 read sequencing and de novo assembly of plant genomes. *Trends Plant Sci*. 2019;24:700-724.
- 6 3. Jung H, Ventura T, Chung J, Kim WJ, Nam BH, Kong H, et al. Twelve quick steps for
7 genome assembly and annotation in the classroom. *PLoS Comput Biol*. 2020;16:e1008325.
- 8 4. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. *Proc Natl*
9 *Acad Sci USA*. 1988;85:2444–2448.
- 10 5. Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM. The Sanger FASTQ file format for
11 sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res*.
12 2010;38:1767–1771.
- 13 6. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence
14 alignment/map format and SAMtools. *Bioinformatics*. 2009;25:2078–2079.
- 15 7. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, et al. The human
16 genome browser at UCSC. *Genome Res*. 2002;12:996–1006.
- 17 8. Pertea G, Pertea M. GFF Utilities: GffRead and GffCompare. *F1000Research*. 2020;9:304.
- 18 9. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, et al. The variant call
19 format and VCFtools. *Bioinformatics*. 2011;27:2156–2158.
- 20 10. Kim DW, Kim RN, Kim DS, Choi SH, Chae SH, Park HS. easySEARCH: A user-
21 friendly bioinformatics program that enables BLAST searching with a massive number of
22 query sequences. *Bioinformation*. 2012;8:792–794.
- 23 11. Blanco-Míguez A, Fdez-Riverola F, Sánchez B, Lourenço A. BlasterJS: A novel
24 interactive JavaScript visualisation component for BLAST alignment results. *PLoS One*.
25 2018;13:e0205286.
- 26 12. Priyam A, Woodcroft BJ, Rai V, Moghul I, Munagala A, Ter F, et al. Sequenceserver: A
27 Modern Graphical User Interface for Custom BLAST Databases. *Mol Biol Evol*.
28 2019;36:2922–2924.
- 29 13. Singh U, Wurtele, ES. "orfipy: a fast and flexible tool for extracting ORFs". *Bioinformatics*.
30 2021;btab090.
- 31 14. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. *Gigascience*.
32 2021;10:giab008.
- 33 15. Baker QB, Hammad MM, Al-Rashdan W, Jararweh Y, Al-Smadi M, Al-Zinati M.
34 Comprehensive comparison of Cloud-based NGS data analysis and alignment tools.
35 *Informatics in Medicine Unlocked*. 2020;18:100296.

- 1 16. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, et al. The Galaxy
2 platform for accessible, reproducible and collaborative biomedical analyses: 2018 update.
3 *Nucleic Acids Res.* 2018;46:W537–W544.
- 4 17. Serrano-Solano B, Föll MC, Gallardo-Alba C, Erxleben A, Rasche H, Hiltemann S, et al.
5 Fostering accessible online education using Galaxy as an e-learning platform. *PLoS Comp Biol.*
6 2021;17:e1008923.
- 7 18. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely
8 available Python tools for computational molecular biology and bioinformatics.
9 *Bioinformatics.* 2009;25:1422–1423.
- 10 19. Du L, Liu Q, Fan Z, Tang J, Zhang X, Price M, et al. Pyfastx: a robust Python package
11 for fast random access to sequences from plain and gzipped FASTA/Q files. *Brief Bioinform.*
12 2021; 22:bbaa368.
- 13 20. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+:
14 architecture and applications. *BMC Bioinformatics.* 2009;10:421.
- 15 21. Kent, WJ. BLAT--the BLAST-like alignment tool. *Genome Res.* 2002;12):656–664.
- 16 22. Altschul S. Gapped BLAST and PSI-BLAST: a new generation of protein database
17 search programs. *Nucleic Acids Res.* 1997;25: 3389–3402.
- 18 23. Bhagwat M, Young L, Robison R. Using BLAT to find sequence similarity in closely
19 related genomes. *Curr Protoc Bioinformatics.* 2012;Chapter10:Unit10.8.
- 20 24. Deonier R, Tavaré S, Waterman M. *Computational Genome Analysis: an introduction.*
21 Springer-Verlag. 2005;25.
- 22 25. Dwivedi VD, Mishra SK. ORF Investigator: A New ORF finding tool combining Pairwise
23 Global Gene Alignment. *Res J Recent Sci.* 2012;1:32–35.
- 24 26. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic
25 features. *Bioinformatics.* 2010;26:841–842.
- 26 27. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. *Nat*
27 *Methods.* 2015;12:59–60.
- 28 28. Buchfink B, Reuter K, Drost HG. Sensitive protein alignments at tree-of-life scale using
29 DIAMOND. *Nat Methods.* 2021;18:366–368.
- 30 29. Wang M, Kong L. pblat: a multithread blat algorithm speeding up aligning sequences to
31 genomes. *BMC Bioinformatics.* 2019;20:28.

32

It would take several seconds to read and write data. Please be patient until you see "OK" dialogue box for each step.

Database Blast Run Result Convert XML Parsing

Input File Database Type Browse

▼ More Options Manually

Out Folder Browse

Database Name

Create

A)

Project Folder Help Log

Project Folder Browse Set Project

Name	Size	Type
TN2TFQ_RNS1.fasta.fxi	21.95 MB	fxi File
TN2TFQ_RNS1.fasta	109.67 MB	fasta File
TN2TFQ_2.fa	291 bytes	fa File
SPD_CN1K_CtgRN.fasta.fxi	12.14 MB	fxi File
SPD_CN1K_CtgRN.fasta	3.75 GiB	fasta File
Small_Query.fasta.fxi		fxi File
Small_Query.fasta		fasta File
GMAP_TN2_Orig.gtf		gtf File
GMAP_TN2_Orig.gtf3		gtf3 File

B)

Fasta Information

Type: DNA

Total Sequence(bp) : 3,988,946,937

Sequence Counts : 116,288

Sequence Average Length(bp) : 34,302.3092408503

Sequence Median Length(bp) : 24,711.0

GC Content(%) : 35.86

N50 and L50(bp) : (17676, 80054)

N75 and L75(bp) : (26711, 52833)

N90 and L90(bp) : (17676, 80054)

OK

C)

easyfm

File Tools View Help

Project Folder Help Log

easyfm

easyfm has four main tools: *Blast*, *Blat*, *ORF* and *File Manipulation*.

Blast

[BLAST® Command Line Applications User Manual](#)

- Create database - [command options](#)
- Run Blast - [command options](#)
 - [blastn](#)
 - [blastx](#)
 - [tblastn](#)
 - [tblastx](#)
- Convert Blast result - [command options](#)
- Parse Blast XML result

Blat

[What is Blat?](#)

[Run Blat - command options](#)

D)

Project Folder Help Log

Command *****

```
C:\EasyFM_Taek\easyfm-v082\easyfm-v082\bin\pbi-blast\exes\makeblastdb.exe -i C:\EasyFM_Taek\EasyFM_Test_Data\SPD_CN1K_CtgRN.fasta -dtype nul
```

Building a new DB, current time: 09/02/2021 12:05:14

New DB name: C:\EasyFM_Taek\EasyFM_Test_Data\SPD_CN1K_CtgRN.fasta

New DB title: C:\EasyFM_Taek\EasyFM_Test_Data\SPD_CN1K_CtgRN.fasta

Sequence type: Nucleotide

Keep MBits: T

Maximum file size: 1000000000B

D) Adding sequences from FASTA; added 116288 sequences in 20.7673 seconds.

Database Blast Run Result Convert XML Parsing

Input File Database Type Database Type Database Type Nucleotide Protein

Out Folder

Database Name

A)

Information

Database is created.

B)

Project Folder

Name	Size	Type
<input type="checkbox"/> TrNASM_2TFQ_RNS1.fasta.fxi	21.95 MB	fxi File
<input type="checkbox"/> TrNASM_2TFQ_RNS1.fasta	109.67 MB	fasta File
<input checked="" type="checkbox"/> TrN2TFQ_2.fa	291 bytes	fa File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.nto	454.25 KB	nto File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.ntf	488.28 KB	ntf File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.nsq	951.11 MB	nsq File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.not	1.33 MB	not File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.nin	1.33 MB	nin File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.nhr	8.85 MB	nhr File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.ndb	488.28 KB	ndb File
<input type="checkbox"/> SPD_CN1K_CtgRN.fasta.fxi	12.14 MB	fxi File
<input checked="" type="checkbox"/> SPD_CN1K_CtgRN.fasta	3.75 GB	fasta File

B)

It would take several seconds to read and write data. Please be patient until you see "OK" dialogue box for each step.

Database Blast Run Result Convert XML Parsing

Database Name

Input Query File

Out File

blastn 1e-10 5. XML Blast output 1

Raw Options

Processing...

Information Finished.

C)

Database Blast Run Result Convert XML Parsing

This tool generates CSV and alignment files.

Input File

Out File

Query Information

Query-ID Query-Length Query_Start Query_End

Target Information

Target_Length Target_Start Target_End

Matches Information

Match_Length Score Matches_Strain

Cut off Values

Rank ≤ 1 HSP number ≤ 1 E value ≤ 1e-1 Identify(%) ≥ 0 Query Coverage(%) ≥ 0

E)

Database Blast Run Result Convert XML Parsing

ASN(BLAST Archive format) is generated with Blast option 11. Please select an ASN format file and an author format option you want to convert.

Input File

Out File

Output Type

- 0. Pairwise
- 1. Query-anchored showing identities
- 2. Query-anchored no identities
- 3. Flat query-anchored, show identities
- 4. Flat query-anchored, no identities
- 5. XML Blast output
- 6. Tabular
- 7. Tabular with comment lines
- 8. Text ASN.1

Raw Options

D)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Input_query-ID	Query_Length	Rank	Target_Definition	Target_Length	Hsp_num	Score	E-Value	Query_Start	Query_End	Query_Coverage(%)	Target_Start	Target_End	Strand	Identity(%)	Match_Length	Accession
2	TrN2TFQ_1	505	1	SPOONKCT_90342	47008	1	917.059	0	1	505	100	30961	35465	1	99.40594059	505	90342
3	TrN2TFQ_2	279	1	SPOONKCT_101686	54527	1	251.044	2.43E-76	119	279	57.34767025	30346	30505	1	99.37888109	161	101685
4	TrN2TFQ_3	275	1	SPOONKCT_70900	59802	1	455.396	6.01E-126	1	275	98.90909091	50020	50109	0	96.73121070	275	70900
5	TrN2TFQ_13	202	1	SPOONKCT_102566	63084	1	374.044	1.62E-101	1	202	100	49536	49707	1	100	202	102565

EasyFM_Figure2



Light weight only (Recommended to use a small set data. For big genome in Database Name, please use each chromosome file, respectively).

Database Name

Input File

Out File

Database Type

Query Type

Tile Size

Step Size

One Off

▼ More Options Manually

Options

EasyFM_Figure3



Light weight only (Recommended to use a small set data).

Input File

Browse

Nucleotide Out File

Change Folder

Protein Out File

Change Folder

Genetic code

Genetic code

1. The Standard Code
2. The Vertebrate Mitochondrial Code
3. The Yeast Mitochondrial Code
4. The Mold, Protozoan, and Coelenterate Mitochondrial Code and the Mycoplasma/Spiroplasma Code
5. The Invertebrate Mitochondrial Code
6. The Ciliate, Dasydadacean and Hexamita Nuclear Code
9. The Echinoderm and Flatworm Mitochondrial Code
10. The Euplotid Nuclear Code
11. The Bacterial, Archaeal and Plant Plastid Code

Minimum Length

Maximum Length

Find ORF

EasyFM_Figure4

Index/Extract Sequence Reverse Convert FastQ to FastA Extract GFF/GTF

Light weight only (Recommended to use a small set data).

FASTA/FastQ File Browse

Index Output File Change Folder

Save Indexes with User's File

Please insert your prefix ID list as a single .txt file without '>' sign that you want to extract and save.

Input User's Index File Browse

Save Indexes with Prefix

Please make sure your original prefix ID is concise without any spaces or special characters. And then, type the unique ID(s) that you want to extract and save.

Prefix ID Save Indexes

Please double click an item to extract and save.

Index

- SPDCN1KCT_1
- A)** SPDCN1KCT_2
- SPDCN1KCT_3

Index/Extract Sequence Reverse Convert FastQ to FastA Extract GFF/GTF

Fasta File Browse

Output File Change Folder

Select a Folder to reverse

Input Folder Set Folder

Output Folder Set Folder

B) Reverse Complement Reverse

Index/Extract Sequence Reverse Convert FastQ to FastA Extract GFF/GTF

Light weight only (Recommended to use a small set data).

Query FastQ File Browse

Out FastA File Change Folder

C) Extract

Index/Extract Sequence Reverse Convert FastQ to FastA Extract GFF/GTF

GFF/GTF File Browse

Reference Fasta File Browse

Filtering Options

D) 483766 Feature Type Filter Save all filtered features

Index/Extract Sequence Reverse Convert FastQ to FastA Extract GFF/GTF

GFF/GTF File Browse

Reference Fasta File Browse

Filtering Options

Count: 5399981 Feature Type Filter Save all filtered features

Please double click an item to save.

	Sequence Name	Source	Feature Type	Score	Strand	Frame			
0	SPDCN1KCT_90242	T1N2_GMAP_DB	gene	38961	39465	.	+	.	ID= T1N2TFQ_1.path1;Name= T1N2TFQ_1;Ori= indetermi
1	SPDCN1KCT_90242	T1N2_GMAP_DB	mRNA	38961	39465	.	+	.	ID= T1N2TFQ_1.mrna1;Name= T1N2TFQ_1;Parent= T1N2TFQ
2	SPDCN1KCT_90242	T1N2_GMAP_DB	exon	38961	39465	99	+	.	ID= T1N2TFQ_1.mrna1.exon1;Name= T1N2TFQ_1;Parent= T1
D) 3	SPDCN1KCT_90242	T1N2_GMAP_DB	CDS	38963	39115	100	+	0	ID= T1N2TFQ_1.mrna1.cds1;Name= T1N2TFQ_1;Parent= T1

EasyFM_Figure5