Open access • Posted Content • DOI:10.1101/2021.09.29.462291

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

Jung H, Jeon B, Daniel Ortiz-Barrientos

Institutions: University of Queensland

Published on: 30 Sep 2021 - bioRxiv (Cold Spring Harbor Laboratory)

Topics: File format, Executable, Graphical user interface, Software suite and Window (computing)

easyfm: An **easy** software suite for **fi**le **m**anipulation of Next Generation Sequencing data on desktops

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

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

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

[a] Equal contributions

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

Abstract

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

Author summary

The analysis and manipulation of NGS data for understanding biological phenomena is an increasingly important aspect in the life sciences. Yet, most methods for analysing, storing and manipulating NGS data require complex command-line tools in HPC or web-based servers and have not yet been implemented in comprehensive, easy-to-use software. This is a major hurdle preventing more general application in the field of NGS data analysis and file manipulation. Here we present *easyfm*, 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. For user-friendliness and convenience, *easyfm* was developed with four work modules and a secondary GUI window (herein secondary window), covering different aspects of NGS data analysis (mainly focusing on FASTA files), 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 (hosted at https://github.com/TaekAndBrendan/easyfm) including a comprehensive step-by-step guide.

1 Introduction

With the broad implementation of NGS technologies in the life sciences, genomics and transcriptomics sequencing data are generated at an unprecedented rate [1–3]. 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 [2,3]. 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 [1,3]. 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 [4] and FASTQ [5] file formats are generated by most NGS platforms, and further SAM/BAM [6], BED [7], GFF/GTF [8], and VCF [9] 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 [10]; BlasterJS [11]; Sequenceserver [12]; orfipy [13]); Samtools and BCFtools [14] 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 [15]. 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 [16,17]. 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 Design and implementation

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 (Fig 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.

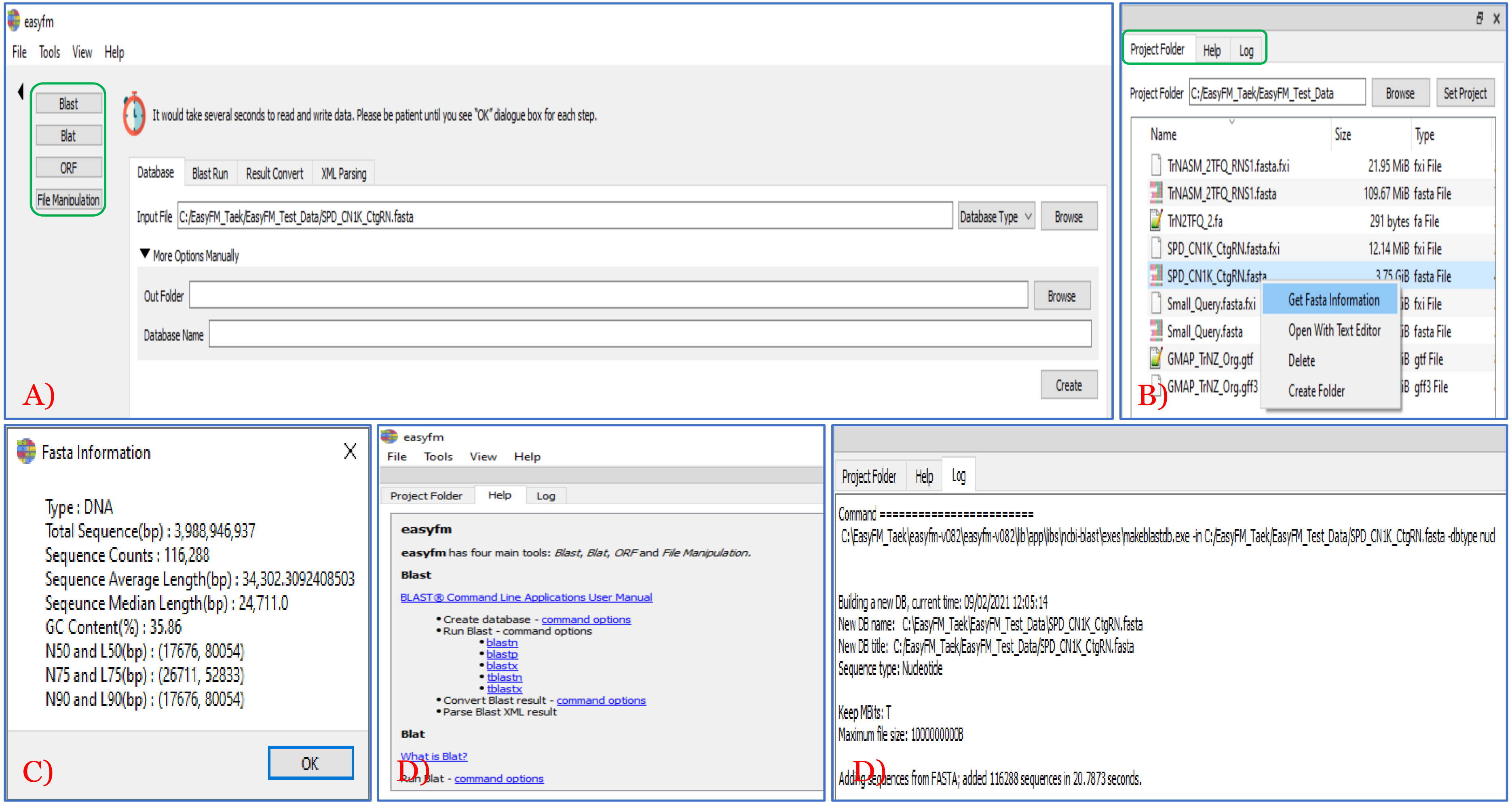
Software	Application
Python Packages	
Biopython [18]	Biopython is a set of freely available tools for common bioinformatics tasks including biological computation.
PyQt5 (https://pypi.org/project/PyQt5/)	PyQt is a Python binding of the cross-platform GUI toolkit Qt.
gffutils (https://github.com/daler/gffutils)	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.

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 Fig 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 Fig 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 (Fig 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 (Fig 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 (Fig 1D). This can aid in effective communication when reporting and resolving any program issues.



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

Input File C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Project Folder C:/EasyFM_Taek/EasyFM_Test_Data

Name	Size	Type
TrNASM_2TFQ_RNS1.fasta.fxi	21.95 MiB	fxi File
TrNASM_2TFQ_RNS1.fasta	109.67 MiB	fasta File
TrN2TFQ_2.fa	291 bytes	fa File
SPD_CN1K_CtgRN.fasta.fxi	12.14 MiB	fxi File
SPD_CN1K_CtgRN.fasta	3.75 GiB	fasta File
Small_Query.fasta.fxi	iB	fxi File
Small_Query.fasta	iB	fasta File
GMAP_TrNZ_Org.gtf	iB	gtf File
GMAP_TrNZ_Org.gff3	iB	gff3 File

Fasta Information

Type : DNA
Total Sequence(bp) : 3,988,946,937
Sequence Counts : 116,288
Sequence Average Length(bp) : 34,302.3092408503
Seqeunce 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)

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
    - blastp
    - blastx
    - tblastn
    - tblastx
- Convert Blast result - command options
- Parse Blast XML result

Blat

What is Blat?

Run Blat - command options

```
Command ========================
 C:\EasyFM_Taek\easyfm-v082\easyfm-v082\lib\app\libs\ncbi-blast\exes\makeblastdb.exe -in C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta -dbtype nucl



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

Adding sequences from FASTA; added 116288 sequences in 20.7873 seconds.
```

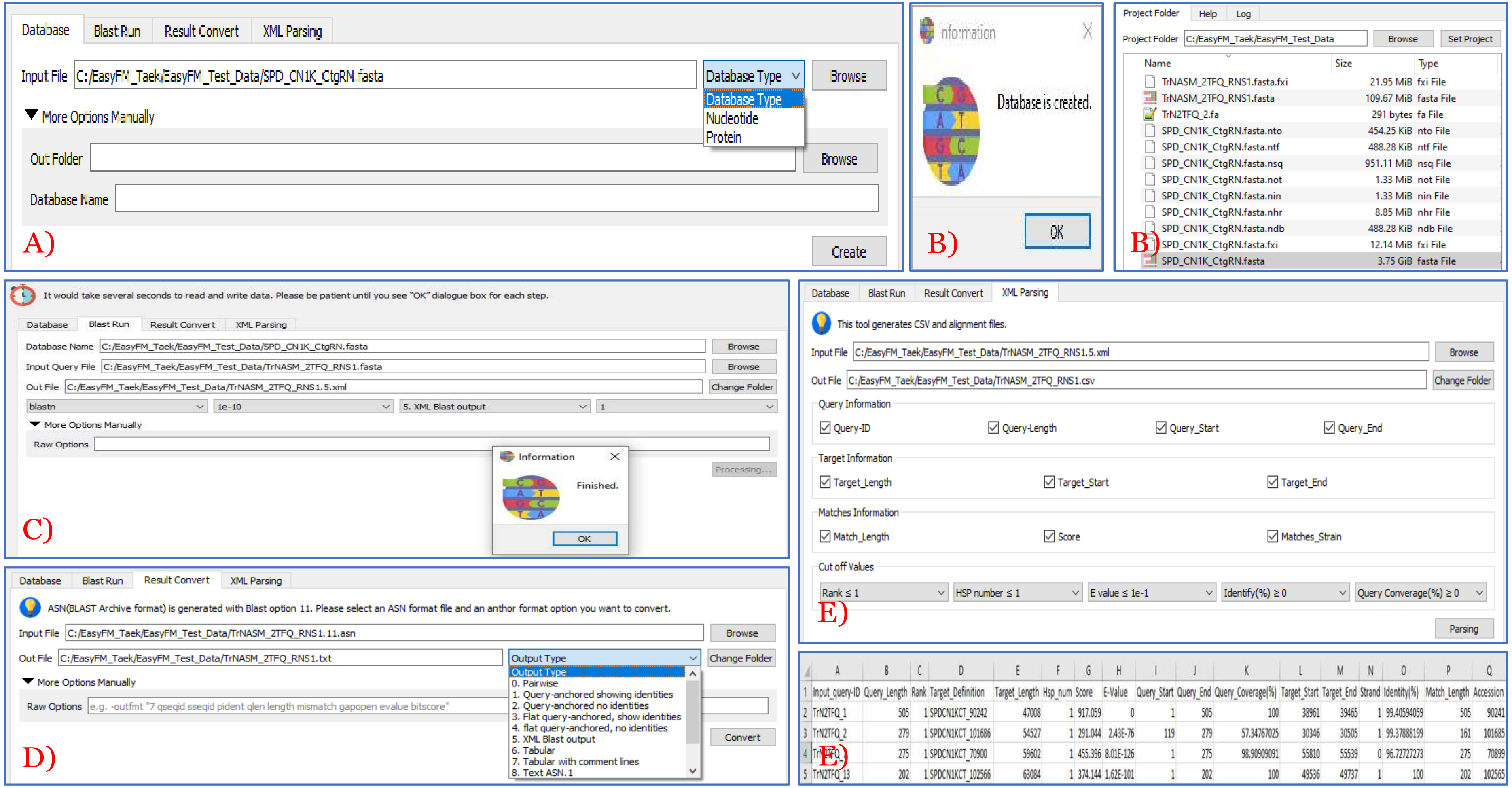
Fig 1. Integration of secondary window with main work module of *easyfm*. A) Four main work modules (green box) to BLAST, BLAT, ORF, and File Manipulation. B) Three secondary modules (green box) to assist with main work modules and extra features using a right mouse click. C) Fasta file stats information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

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 [18,22]. 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 (Fig 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 (Fig 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 (Fig 2E). Along with recent free tools [10–12], *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.



A)

Input File C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Out Folder

Database Name

B)

Database is created.

Project Folder C:/EasyFM_Taek/EasyFM_Test_Data

Name	Size	Type
TrNASM_2TFQ_RNS1.fasta.fxi	21.95 MiB	fxi File
TrNASM_2TFQ_RNS1.fasta	109.67 MiB	fasta File
TrN2TFQ_2.fa	291 bytes	fa File
SPD_CN1K_CtgRN.fasta.nto	454.25 KiB	nto File
SPD_CN1K_CtgRN.fasta.ntf	488.28 KiB	ntf File
SPD_CN1K_CtgRN.fasta.nsq	951.11 MiB	nsq File
SPD_CN1K_CtgRN.fasta.not	1.33 MiB	not File
SPD_CN1K_CtgRN.fasta.nin	1.33 MiB	nin File
SPD_CN1K_CtgRN.fasta.nhr	8.85 MiB	nhr File
SPD_CN1K_CtgRN.fasta.ndb	488.28 KiB	ndb File
SPD_CN1K_CtgRN.fasta.fxi	12.14 MiB	fxi File
SPD_CN1K_CtgRN.fasta	3.75 GiB	fasta File

C)

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

Database Name C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Input Query File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.fasta

Out File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.5.xml

blastn | 1e-10 | 5. XML Blast output | 1

Raw Options

Finished.

D)

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

Input File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.11.asn

Out File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.txt

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 e.g. -outfmt "7 qseqid sseqid pident qlen length mismatch gapopen evalue bitscore"

E)

This tool generates CSV and alignment files.

Input File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.5.xml

Out File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.csv

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

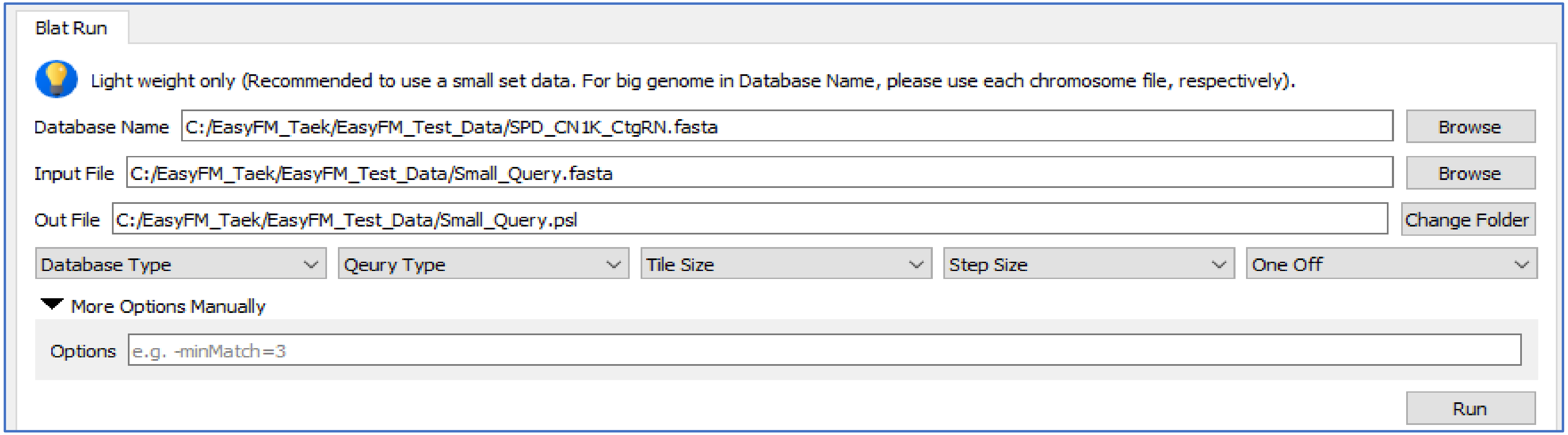
Cut off Values
Rank ≤ 1 | HSP number ≤ 1 | E value ≤ 1e-1 | Identify(%) ≥ 0 | Query Converage(%) ≥ 0

	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	SPDCN1KCT_90242	47008	1	917.059	0	1	505	100	38961	39465	1	99.40594059	505	90241
3	TrN2TFQ_2	279	1	SPDCN1KCT_101686	54527	1	291.044	2.43E-76	119	279	57.34767025	30346	30505	1	99.37888199	161	101685
4	TrN2TFQ_[illegible]	275	1	SPDCN1KCT_70900	59602	1	455.396	8.01E-126	1	275	98.90909091	55810	55539	0	96.72727273	275	70899
5	TrN2TFQ_13	202	1	SPDCN1KCT_102566	63084	1	374.144	1.62E-101	1	202	100	49536	49737	1	100	202	102565

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.

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 [21]. While BLAT is less sensitive than BLAST, BLAT has a few clear advantages over BLAST from a practical standpoint in speed and convenience [23]. 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 [21]. 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 [23], but not a user-friendly GUI. However, *easyfm* BLAT (v3.2.1) enables users to control all parameters with a simple mouse click (Fig 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.



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

Database Name C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

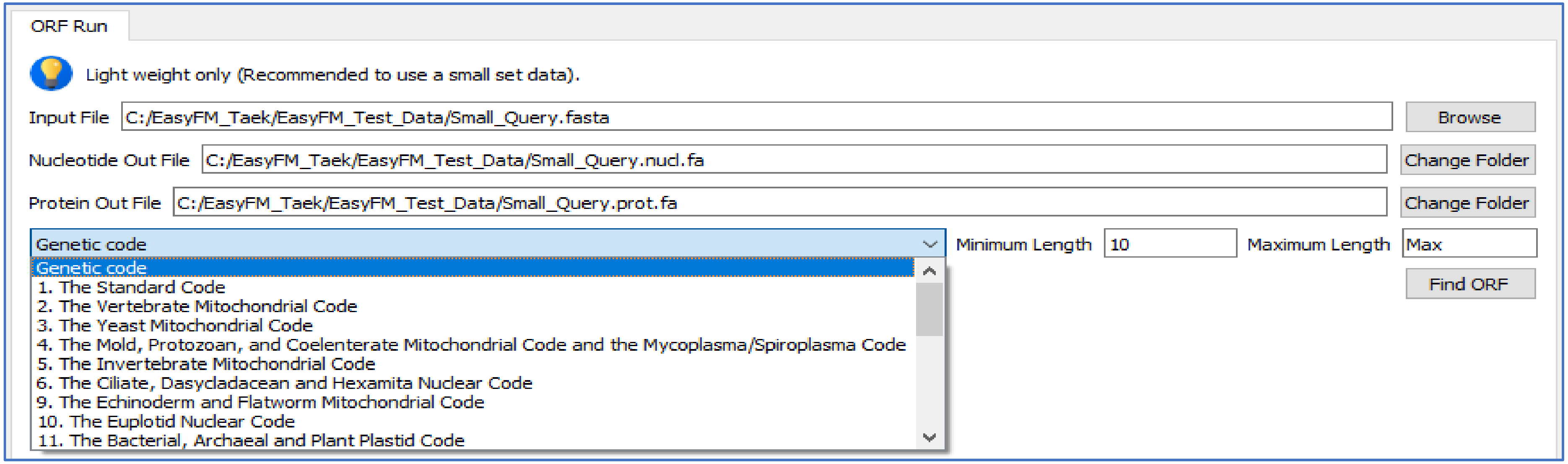
Input File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.fasta

Out File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.psl

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.

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 [24]. 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 [25] and orfipy [13]) 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) (Fig 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 [13,25], *easyfm* ORF will provide rapid, flexible searches in multiple output formats to allow the easy downstream analysis of ORFs.



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

Input File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.fasta

Nucleotide Out File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.nucl.fa

Protein Out File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.prot.fa

Minimum Length 10

Maximum Length Max

Browse
Change Folder
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, Dasycladacean and Hexamita Nuclear Code
9. The Echinoderm and Flatworm Mitochondrial Code
10. The Euplotid Nuclear Code
11. The Bacterial, Archaeal and Plant Plastid Code
Find ORF

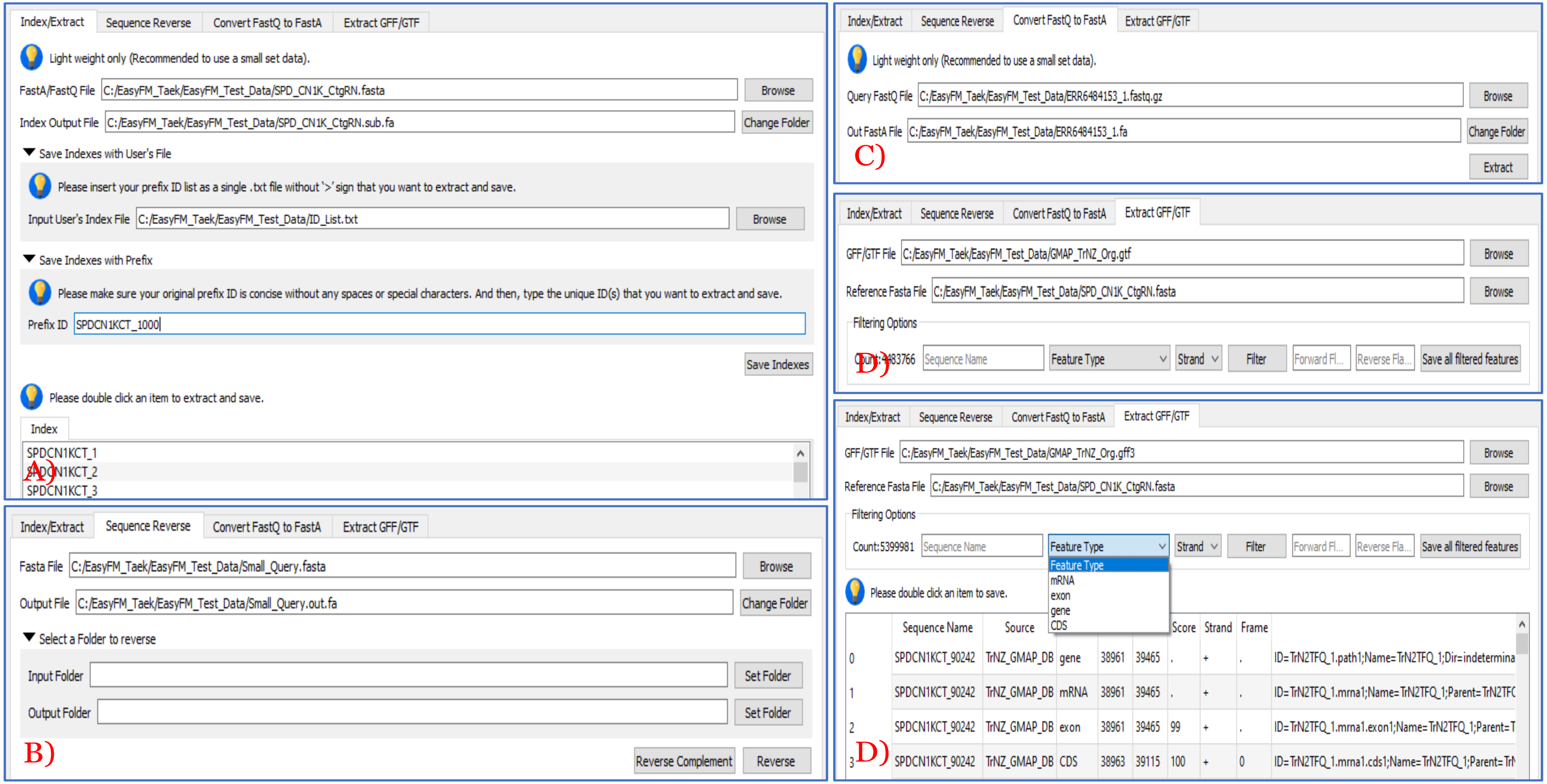
Fig 4. User-friendly standalone work modules in *easyfm*: ORF module. Most steps include further options for a user-specified parameter. Run ORF with different genetic codes for coding and protein sequences. A FASTA format output file of nucleotide and protein from a six-frame translation will be generated.

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 [14]; BEDTools [26]) 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) (Fig 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 (Fig 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 (Fig 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 (Fig 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 [14,26], *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.



A)

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

FastA/FastQ File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Index Output File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.sub.fa

▼ 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: C:/EasyFM_Taek/EasyFM_Test_Data/ID_List.txt

▼ 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: SPDCN1KCT_1000

Please double click an item to extract and save.

SPDCN1KCT_1
SPDCN1KCT_2
SPDCN1KCT_3

B)

Fasta File: C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.fasta

Output File: C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.out.fa

▼ Select a Folder to reverse

Input Folder:

Output Folder:

C)

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

Query FastQ File: C:/EasyFM_Taek/EasyFM_Test_Data/ERR6484153_1.fastq.gz

Out FastA File: C:/EasyFM_Taek/EasyFM_Test_Data/ERR6484153_1.fa

D)

GFF/GTF File: C:/EasyFM_Taek/EasyFM_Test_Data/GMAP_TrNZ_Org.gtf

Reference Fasta File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Filtering Options

Count: 483766 | Sequence Name | Feature Type | Strand | Forward Fl... | Reverse Fla...

GFF/GTF File: C:/EasyFM_Taek/EasyFM_Test_Data/GMAP_TrNZ_Org.gff3

Reference Fasta File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Filtering Options

Count:5399981 | Sequence Name | Feature Type | Strand | Forward Fl... | Reverse Fla...

Feature Type
mRNA
exon
gene
CDS

Please double click an item to save.

	Sequence Name	Source				Score	Strand	Frame	
0	SPDCN1KCT_90242	TrNZ_GMAP_DB	gene	38961	39465	.	+	.	ID=TrN2TFQ_1.path1;Name=TrN2TFQ_1;Dir=indetermina
1	SPDCN1KCT_90242	TrNZ_GMAP_DB	mRNA	38961	39465	.	+	.	ID=TrN2TFQ_1.mrna1;Name=TrN2TFQ_1;Parent=TrN2TFC
2	SPDCN1KCT_90242	TrNZ_GMAP_DB	exon	38961	39465	99	+	.	ID=TrN2TFQ_1.mrna1.exon1;Name=TrN2TFQ_1;Parent=T
3	SPDCN1KCT_90242	TrNZ_GMAP_DB	CDS	38963	39115	100	+	0	ID=TrN2TFQ_1.mrna1.cds1;Name=TrN2TFQ_1;Parent=Tr

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.

Availability and future directions

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 [27,28], (Buchfink et al. 2015, 2021) and pBLAT [29] 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 their colleagues (special thanks to Maddie James, Melanie Wilkinson, Cara Conradsen, and Nicholas O'Brien) and collaborators for their valuable and constructive comments.

References

1. Breese MR, Liu Y. NGSUtils: a software suite for analyzing and manipulating next-generation sequencing datasets. Bioinformatics. 2013;29:494–496.

2. Jung H, Winefield C, Bombarely A, Prentis P, Waterhouse, P. Tools and strategies for long-read sequencing and de novo assembly of plant genomes. Trends Plant Sci. 2019;24:700-724.

3. Jung H, Ventura T, Chung J, Kim WJ, Nam BH, Kong H, et al. Twelve quick steps for genome assembly and annotation in the classroom. PLoS Comput Biol. 2020;16:e1008325.

4. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci USA. 1988;85:2444–2448.

5. Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res. 2010;38:1767–1771.

6. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence alignment/map format and SAMtools. Bioinformatics. 2009;25:2078–2079.

7. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, et al. The human genome browser at UCSC. Genome Res. 2002;12:996–1006.

8. Pertea G, Pertea M. GFF Utilities: GffRead and GffCompare. F1000Research. 2020;9:304.

9. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, et al. The variant call format and VCFtools. Bioinformatics. 2011;27:2156–2158.

10. Kim DW, Kim RN, Kim DS, Choi SH, Chae SH, Park HS. easySEARCH: A user-friendly bioinformatics program that enables BLAST searching with a massive number of query sequences. Bioinformation. 2012;8:792–794.

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

12. Priyam A, Woodcroft BJ, Rai V, Moghul I, Munagala A, Ter F, et al. Sequenceserver: A Modern Graphical User Interface for Custom BLAST Databases. Mol Biol Evol. 2019;36:2922–2924.

13. Singh U, Wurtele, ES. "orfipy: a fast and flexible tool for extracting ORFs". Bioinformatics. 2021;btab090.

14. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. Gigascience. 2021;10:giab008.

15. Baker QB, Hammad MM, Al-Rashdan W, Jararweh Y, Al-Smadi M, Al-Zinati M. Comprehensive comparison of Cloud-based NGS data analysis and alignment tools. Informatics in Medicine Unlocked. 2020;18:100296.

16. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46:W537–W544.

17. Serrano-Solano B, Föll MC, Gallardo-Alba C, Erxleben A, Rasche H, Hiltemann S, et al. Fostering accessible online education using Galaxy as an e-learning platform. PLoS Comp Biol. 2021;17:e1008923.

18. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25:1422–1423.

19. Du L, Liu Q, Fan Z, Tang J, Zhang X, Price M, et al. Pyfastx: a robust Python package for fast random access to sequences from plain and gzipped FASTA/Q files. Brief Bioinform. 2021; 22:bbaa368.

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

21. Kent, WJ. BLAT--the BLAST-like alignment tool. Genome Res. 2002;12):656–664.

22. Altschul S. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25: 3389–3402.

23. Bhagwat M, Young L, Robison R. Using BLAT to find sequence similarity in closely related genomes. Curr Protoc Bioinformatics. 2012;Chapter10:Unit10.8.

24. Deonier R, Tavaré S, Waterman M. Computational Genome Analysis: an introduction. Springer-Verlag. 2005;25.

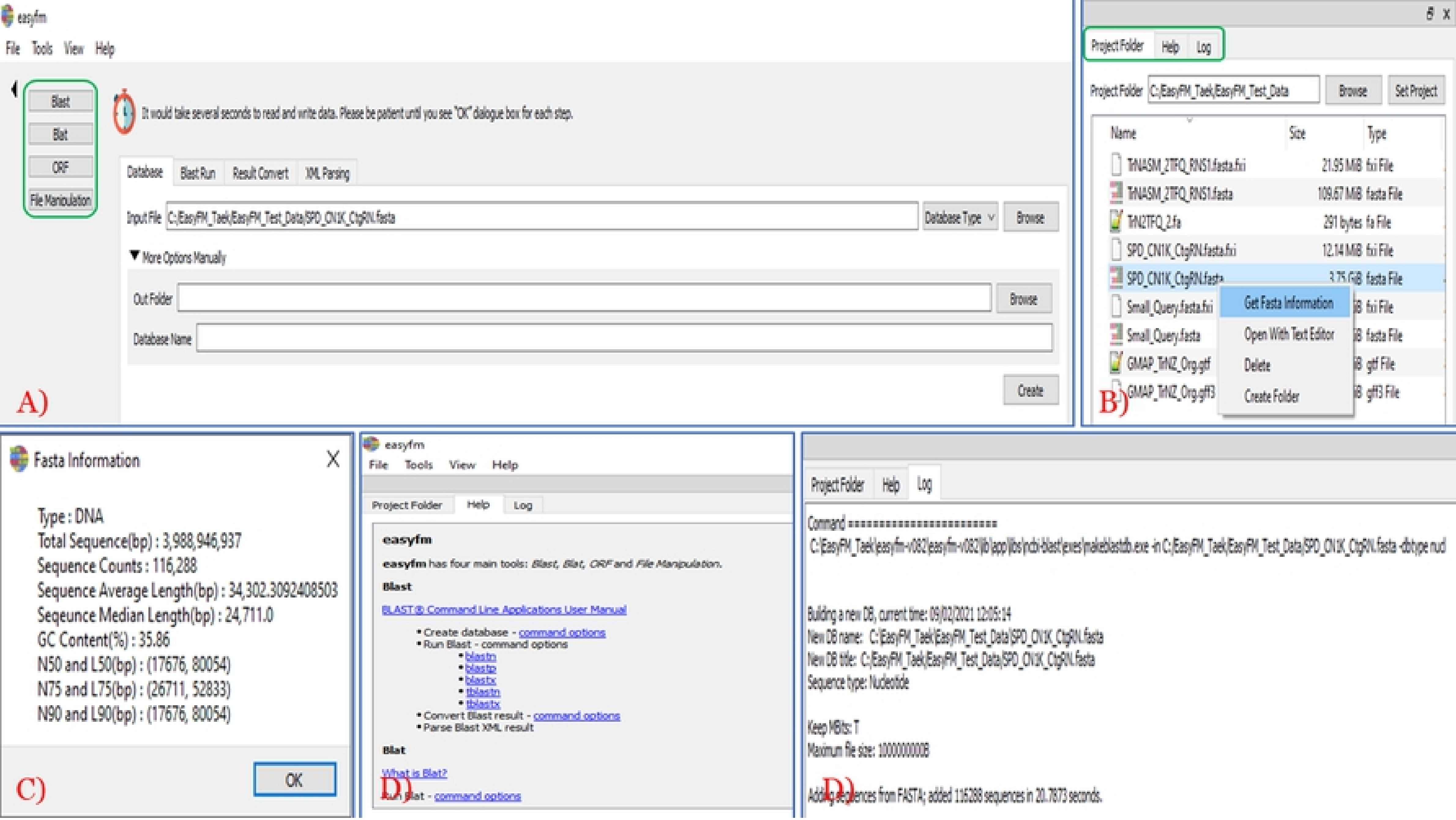
25. Dwivedi VD, Mishra SK. ORF Investigator: A New ORF finding tool combining Pairwise Global Gene Alignment. Res J Recent Sci. 2012;1:32–35.

26. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–842.

27. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Methods. 2015;12:59–60.

28. Buchfink B, Reuter K, Drost HG. Sensitive protein alignments at tree-of-life scale using DIAMOND. Nat Methods. 2021;18:366–368.

29. Wang M, Kong L. pblat: a multithread blat algorithm speeding up aligning sequences to genomes. BMC Bioinformatics. 2019;20:28.



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

Input File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Out Folder:

Database Name:

Project Folder: C:/EasyFM_Taek/EasyFM_Test_Data

Name	Size	Type
TrNASM_2TFQ_RNS1.fasta.fai	21.95 MiB	fai File
TrNASM_2TFQ_RNS1.fasta	109.67 MiB	fasta File
TrN2TFQ_2.fa	291 bytes	fa File
SPD_CN1K_CtgRN.fasta.fai	12.14 MiB	fai File
SPD_CN1K_CtgRN.fasta	3.75 GiB	fasta File
Small_Query.fasta.fai	B	fai File
Small_Query.fasta	B	fasta File
GMAP_TrNZ_Org.gtf	B	gtf File
GMAP_TrNZ_Org.gff3	B	gff3 File

Fasta Information

Type : DNA
Total Sequence(bp) : 3,988,946,937
Sequence Counts : 116,288
Sequence Average Length(bp) : 34,302.3092408503
Seqeunce 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

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
  - blastp
  - blastx
  - tblastn
  - tblastx
- Convert Blast result - command options
- Parse Blast XML result

Blat

What is Blat?

- Run Blat - command options

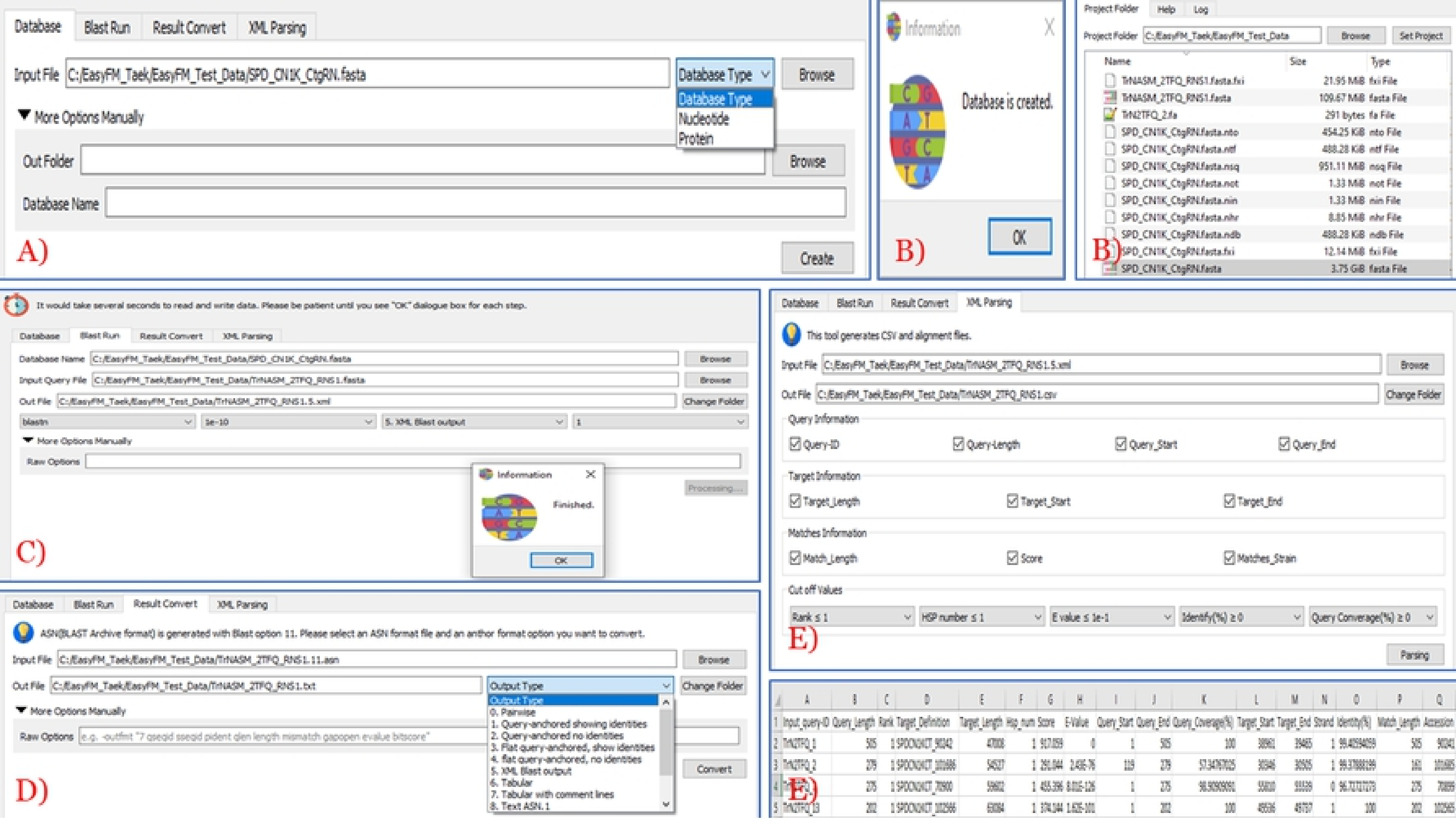
```
Command ==========================
 C:\EasyFM_Taek\easyfm-v082\easyfm-v082\lib\app\libs\ncbi-blast\exes\makeblastdb.exe -in C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta -dbtype nucl


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

Adding sequences from FASTA; added 116288 sequences in 20.7873 seconds.
```

EasyFM_Figure1



Database | Blast Run | Result Convert | XML Parsing

Input File C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Database Type: Database Type / Nucleotide / Protein

▼ More Options Manually

Out Folder

Database Name

Create

Information

Database is created.

OK

Project Folder | Help | Log

Project Folder C:/EasyFM_Taek/EasyFM_Test_Data

Name	Size	Type
TrNASM_2TFQ_RNS1.fasta.fxi	21.95 MiB	fxi File
TrNASM_2TFQ_RNS1.fasta	109.67 MiB	fasta File
TrN2TFQ_2.fa	291 bytes	fa File
SPD_CN1K_CtgRN.fasta.nto	454.25 KiB	nto File
SPD_CN1K_CtgRN.fasta.ntf	488.28 KiB	ntf File
SPD_CN1K_CtgRN.fasta.nsq	951.11 MiB	nsq File
SPD_CN1K_CtgRN.fasta.not	1.33 MiB	not File
SPD_CN1K_CtgRN.fasta.nin	1.33 MiB	nin File
SPD_CN1K_CtgRN.fasta.nhr	8.85 MiB	nhr File
SPD_CN1K_CtgRN.fasta.ndb	488.28 KiB	ndb File
SPD_CN1K_CtgRN.fasta.fxi	12.14 MiB	fxi File
SPD_CN1K_CtgRN.fasta	3.75 GiB	fasta File

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 C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Input Query File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.fasta

Out File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.5.xml

blastn | 1e-10 | 5. XML Blast output | 1

▼ More Options Manually

Raw Options

Processing...

Information

Finished.

OK

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 anthor format option you want to convert.

Input File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.11.asn

Out File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.txt

Output Type:
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

▼ More Options Manually

Raw Options e.g. -outfmt "7 qseqid sseqid pident qlen length mismatch gapopen evalue bitscore"

Convert

Database | Blast Run | Result Convert | XML Parsing

This tool generates CSV and alignment files.

Input File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.5.xml

Out File C:/EasyFM_Taek/EasyFM_Test_Data/TrNASM_2TFQ_RNS1.csv

Query Information
☑ Query-ID ☑ Query-Length ☑ Query_Start ☑ Query_End

Target Information
☑ Target_Length ☑ Target_Start ☑ Target_End

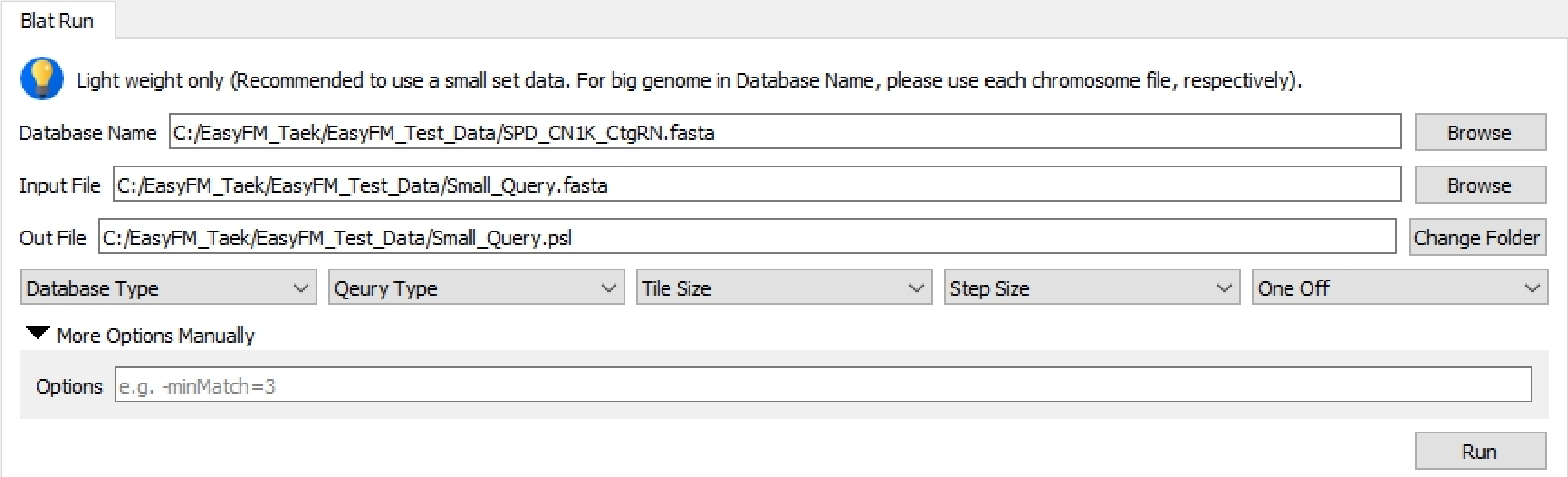
Matches Information
☑ Match_Length ☑ Score ☑ Matches_Strain

Cut off Values
Rank ≤ 1 | HSP number ≤ 1 | E value ≤ 1e-1 | Identify(%) ≥ 0 | Query Converage(%) ≥ 0

Parsing

	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	SPDCN1KCT_9042	47008	1	917.059	0	1	505	100	38961	39465	1	99.40594059	505	9042
3	TrN2TFQ_2	279	1	SPDCN1KCT_101686	54537	1	291.044	2.45E-76	119	279	57.3476703	30346	30505	1	99.3788199	161	101686
4	[illegible]	275	1	SPDCN1KCT_70890	55602	1	455.396	8.01E-126	1	275	98.9090909	55810	55538	0	96.7272727	275	70890
5	TrN2TFQ_13	202	1	SPDCN1KCT_102566	65084	1	374.144	1.62E-101	1	202	100	49536	49737	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).

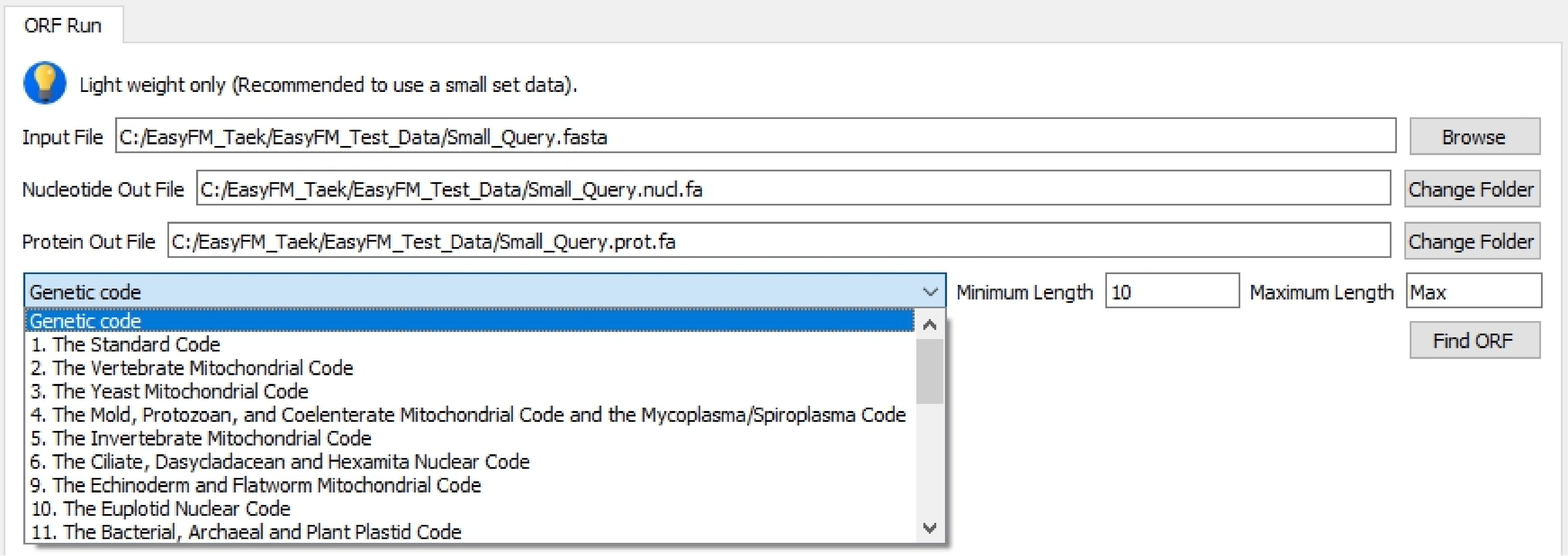
Database Name: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Input File: C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.fasta

Out File: C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.psl

Options: e.g. -minMatch=3

EasyFM_Figure3



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

Input File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.fasta

Nucleotide Out File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.nucl.fa

Protein Out File C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.prot.fa

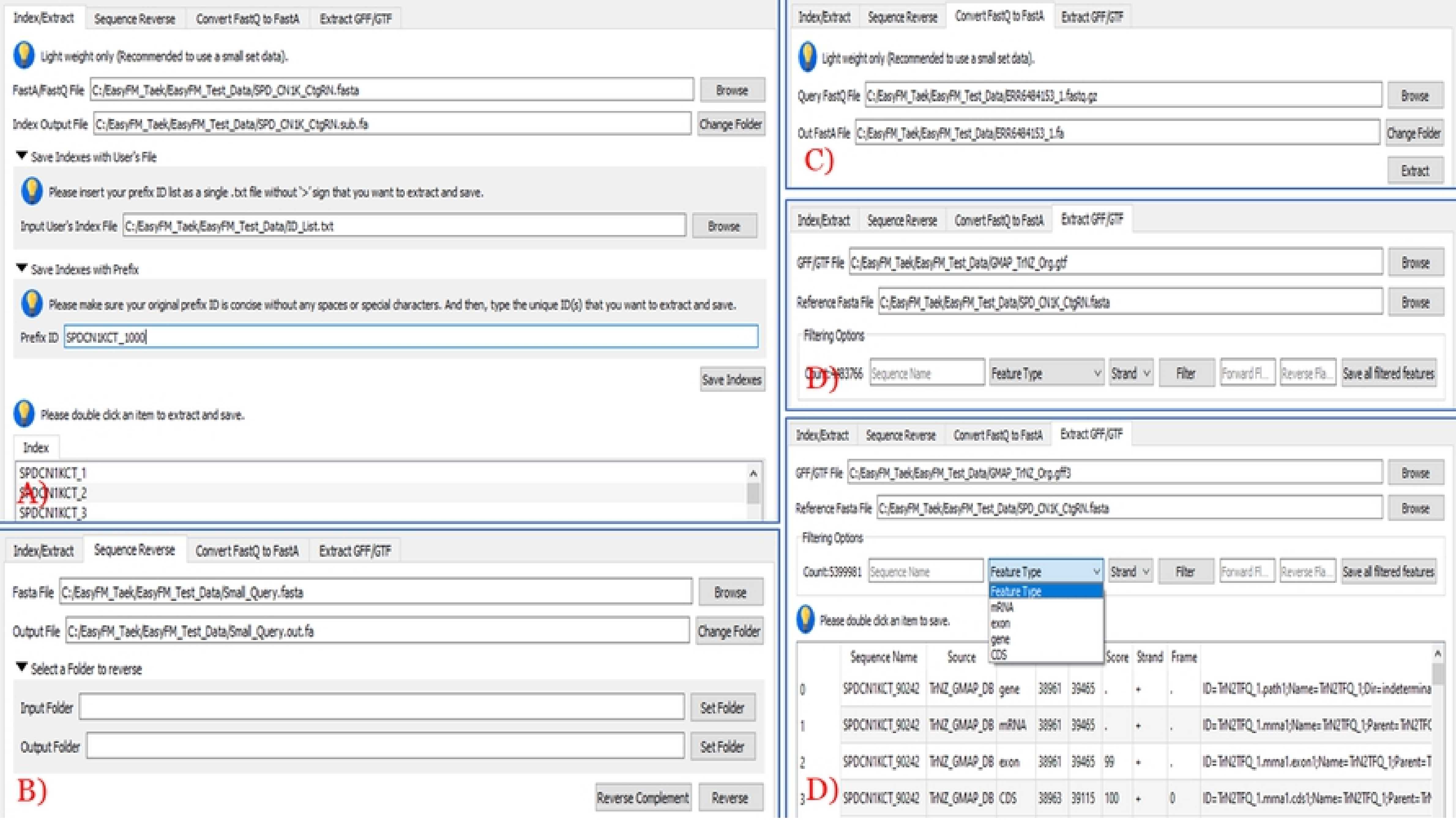
Genetic code

- 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, Dasycladacean 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 10

Maximum Length Max

EasyFM_Figure4



A)

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

FastA/FastQ File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Index Output File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.sub.fa

▼ 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: C:/EasyFM_Taek/EasyFM_Test_Data/ID_List.txt

▼ 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: SPDCN1KCT_1000

Please double click an item to extract and save.

SPDCN1KCT_1
SPDCN1KCT_2
SPDCN1KCT_3

B)

Fasta File: C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.fasta

Output File: C:/EasyFM_Taek/EasyFM_Test_Data/Small_Query.out.fa

▼ Select a Folder to reverse

Input Folder:

Output Folder:

C)

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

Query FastQ File: C:/EasyFM_Taek/EasyFM_Test_Data/ERR6484153_1.fastq.gz

Out FastA File: C:/EasyFM_Taek/EasyFM_Test_Data/ERR6484153_1.fa

D)

GFF/GTF File: C:/EasyFM_Taek/EasyFM_Test_Data/GMAP_TrNZ_Org.gtf

Reference Fasta File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Filtering Options

Count: 483766

D)

GFF/GTF File: C:/EasyFM_Taek/EasyFM_Test_Data/GMAP_TrNZ_Org.gff3

Reference Fasta File: C:/EasyFM_Taek/EasyFM_Test_Data/SPD_CN1K_CtgRN.fasta

Filtering Options

Count: 5399981

Please double click an item to save.

	Sequence Name	Source				Score	Strand	Frame	
0	SPDCN1KCT_90242	TrNZ_GMAP_DB	gene	38961	39465	.	+	.	ID=TrN2TFQ_1.path1;Name=TrN2TFQ_1;Dir=indetermina
1	SPDCN1KCT_90242	TrNZ_GMAP_DB	mRNA	38961	39465	.	+	.	ID=TrN2TFQ_1.mrna1;Name=TrN2TFQ_1;Parent=TrN2TFC
2	SPDCN1KCT_90242	TrNZ_GMAP_DB	exon	38961	39465	99	+	.	ID=TrN2TFQ_1.mrna1.exon1;Name=TrN2TFQ_1;Parent=T
3	SPDCN1KCT_90242	TrNZ_GMAP_DB	CDS	38963	39115	100	+	0	ID=TrN2TFQ_1.mrna1.cds1;Name=TrN2TFQ_1;Parent=Tr

EasyFM_Figure5