FetBASE

Fettuccine Human LINE-1 & HERV Transposable Element Platform

Version 0.1; February 2018

Documentation

Chowdhury Nayam, Grigoriadis Dionysios, Matthews Michaela

What is FetBASE?

FetBASE© - Fettucine Human LINE-1 & HERV Transposable Elements Platform is a fast, reliable and user-friendly interface for exploring Long Interspersed Nuclear Elements 1 (LINE-1) and Human Endogenous Virus (HERV) retrotransposons and their protein expression. This software was developed by "Fettuccine" student group of Queen Mary University of London for the purposes of a software development group project (School of Biological Sciences and Chemistry - MSc Bioinformatics) under the invaluable guidance of Professor Conrad Bessant (https://bessantlab.org/) and Dr Fabrizio Smeraldi (https://goo.gl/k6jxCr).

The software is based on a complete and updated database of all known LINE1 & HERV repeats of the human genome (Genome Reference Consortium Human Build 38 - GRCh38 genome assembly), as well as all the predicted protein sequences that these genomic loci encode.

One part of the platform's tools is responsible for the interactive exploration of the database, as the user can navigate through the tables of LINE1 and HERV families and their translated products, see a visualised distribution of all these elements on the chromosomes and explore their relationships based on their predicted protein sequences. The second part of the tools of the platform focuses on the search of user-input queries against the protein records of the database. More specifically, user can search the entire database with one or multiple protein sequences using a search-box or uploading a file in FASTA format to find out which family member it has been translated from. A direct search of the database with a peptide identification file in mzIdentML or mzTab format is possible, to show and identify if any retrotransposons are being translated in the corresponding.

The resulting information from these uploaded data is being stored in the database to provide an expression atlas of these retrotransposons in human tissue types.

Software architecture

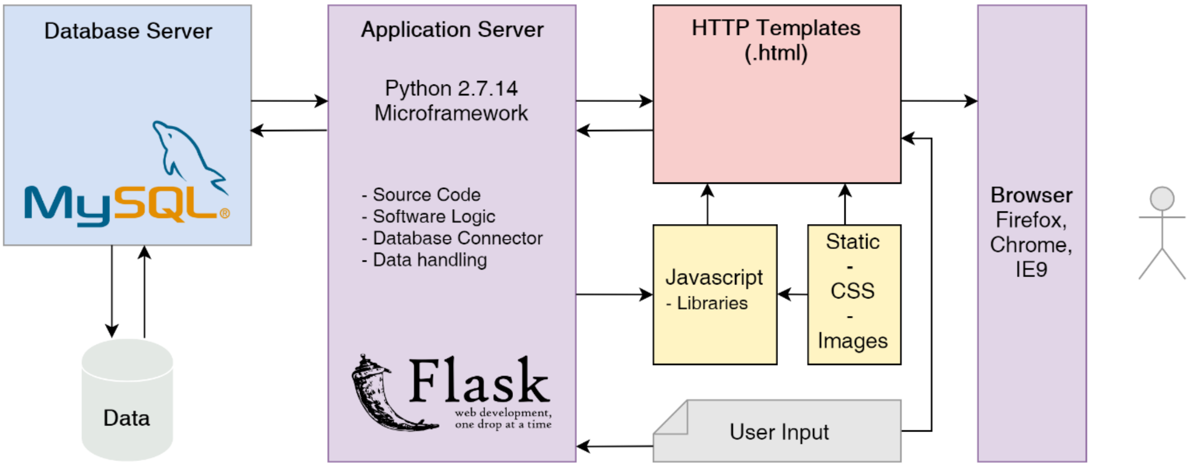

Figure 1. Schematic illustration of the FetBASE software architecture. The platform is based on a MySQL 5.7 database of the retrotransposons and their proteins, which is handled by a MySQL 5.7 database server. The software was developed using Flask Python microframework (Python 2.7.14) which communicates with the database server through mysql Client and runs the source code of the software. Flask communicates and runs along with Javascript to control the behaviour of different objects to the final HTML template. CSS was used to provide style information. The software was tested in Google Chrome, Mozilla Firefox and Internet Explorer 9+ and it was completely functional. Image was created using draw.io.

Packages/Systems Used

Flask

Available at http://flask.pocoo.org/ [1].

Flask is a micro web framework written in python which is also based on the Werkzeug toolkit and uses the Jinja2 template engine which allows for the use of python-like expressions in HTML document. This microframework was chosen as the main web development toolkit as it was quick to learn and understand under the time constraints that were given, which it was matching the developers' knowledge in python 2.7 programming language.

Flask was also used as it, by default, protects against the use of cross site scripting (XSS) so malicious scripts that are injected into the website does not compromise the website security, since software requires and handles input data from the users.

This micro web framework is also compatible with python 3 so that if the website needs to migrate from python 2.7, it can occur smoothly.

The following python 2.7 libraries and modules (other than standard) were used to be able to create the website: Flask, MySQLdb , Pandas (https://pandas.pydata.org), NumPy (http://numpy.org), biopython (http://biopython.org), matplotlib (https://matplotlib.org), Pygraphviz (https://goo.gl/JeZwJR).

MySQL 5.7

Available at http://www.mysql.com [2].

MySQL is an open-source SQL management system, distributed by Oracle Corporation, which was used to add, access and process data stored in FetBASE database. The system is usually known to be fast, reliable and stable and works on all major operating systems, while the fact that it is open-source made it easier to modify it to suit the needs of the software.

Considering the amount of data that FetBASE database contains (millions of protein entries and thousands of retrotransposonal repeats), MySQL Server was a reasonable choice, given the fast and efficient way of querying even on the fly. It was also very simple to use in conjunction with flask through the powerful MySQL Connector/Python and allowed for multiple different data tables to be created and read easily. Thus, MySQL system was a key in reaching a robust performance of the FetBASE software, even in the really potent processes of searching the protein sequences of the database with user-inserted sequences.

MAMP (for windows), AAMP (for linux) was used to run the MySQL server and the web interface (phpmyadmin).

JavaScript libraries

Elements such as professional look combined with smooth functionality and interactive features were of high priority during the software development. For this reason, FetBASE includes and runs some useful JavaScript libraries. For the distribution of the repeats, JQuery was used to create 24 dropdown buttons which allow the user to select and view the distribution for each chromosome. This was done as Jquery allows for the information to be displayed all on one page instead of multiple pages.

Jquery (https://jquery.com):

JQuery is a JavaScript library designed to simplify the client-side scripting of HTML It was used to be able to add JavaScript functionality into the browser. The features that are available with the use of JQuery are endless as it allows you to manipulate and visualise the webpage in many different ways such as adding animations and processing events after certain specifications are met. Jquery plugins that were used:

- DataTables (https://datatables.net/): A Table plugin for JQuery, ideal for projection of large database tables. The main advantage of this plugin is that supports any data source, server-side processing included, something that really enhanced the speed of loading the protein tables of FetBASE database. Its professional quality, beautiful API and easy customization made it first choice.

HTML/CSS

All websites require HTML to be able to run. Combined with CSS and Javascript, the possibilities that can occur are limitless. We opted to go for a minimalistic visual approach to make the website seem more professional and refined.

This was done is varying ways such as using CSS and bootstrapping (Bootstrap v4.0.0 https://getbootstrap.com/). Bootstrapping is a popular free frontend framework which allows for more efficient creation of websites that are visually pleasing, responsive and have the ability to work with devices with smaller screens such as mobile phones.

The use of bootstrapping has reduced the time required to create the website exactly to our specifications by allowing faster and seamless addition of content without having to hand create the CSS manually for every object that is displayed in the website.

FetBASE Database

Getting the Data

Getting the HERV and LINE-1 repeats data

USCS Table browser[3] was used to retrieve all known LINE-1 and HERV repeats of the human genome. More specifically, the table of the human repeat annotations, RepeatMasker version open-3.2.7[4] (rmsk table listed in RepeatMasker track/ Repeats group) was used for USCS genome annotation for the Dec. 2013 assembly of the human genome (hg19, GRCh38 Genome Reference Consortium Human Reference 38). 720,871 LINE-1 sequences and 1,001,410 HERV sequences added to the database with repeat names and HERV classes (named Superfamilies in the Platform) defined based on the Repbase classification system[5].

Prediction of the protein sequences

Based on the obtained genomic sequences, the corresponding protein sequences were predicted by modifying and using the appropriate python script from Rosalind-Problems GitHub repository (https://goo.gl/wm9CHa) which finds Open Reading Frames (ORFs) and translates them. This python code was easy to use and modify while it could identify all ORFs in both directions 5' -> 3' and 3' -> 5' fast and smoothly. For LINE-1 ORFs more than 50 aa where extracted as this is the approximate length of the shortest LINE-1 protein sequence LORF0. For HERVs, only ORFs more than 80 aa where extracted as proposed by So Nakagawa and Mahoko Ueda Takahashi[6].

HMMER 3.1b1 (http://hmmer.org) was used to identify which of these proteins are derived from these retrotransposons. Hmmer is a fast and efficient tool, very easy to use and compatible to HPC systems, among other it can be used to search sequence databases for sequence homologs, and for making sequence alignments.

The above extracted HERV amino acid sequences were searched with hmmsearch tool with viral motif profiles for gag, pol, pro, env and other accessory proteins separately. Hidden Markov Models (HMMs) of these viral motif profiles were downloaded from the Pfam (http://pfam.xfam.org/) and Gypsy databases (http://gydb.org). This method led to the identification of sequences that are convincingly derived from HERV retrotransposons and to an efficient discrimination between gag, pol, pro, env and accessory proteins. Similarly, LORF1 and LORF2 protein sequences from UniprotKB [7] (LORF1: Q9UN81and LORF2: O00370) and LORF0 protein sequence consensus [8] were used to run a jackhammer (PSIBLAST-like) search against the above extracted LINE-1 protein sequences. This method was selected as it performs a fast and efficient iterative search of a protein sequence vs a protein sequence database and it led to the identification of sequences that are convincingly derived from HERV retrotransposons and to an efficient discrimination between lorf0, lorf1 and lorf2 proteins.

Sorting the Data

For each HERV retrotransposon, the HERV class (Superfamily) had already been determined by RepeatMasker and the Family was determined by us using a classification system found in published research work [9], [10]. LINE-1 elements were classified into families based on their repeat names. All the tables that need to be shown in the website, are indexed and predetermined in MySQL, something that significantly reduce the loading times of the software.

Database Schema

MySQL configuration

To efficiently load tables and search against the protein tables with user-input queries, some of the MySQL configuration settings such as key_buffer_size and max_allowed_packet where increased.

Distribution table

This part of the application allows users to select a chromosome to view the distribution of HERV and LINE1 repeats. This includes a dropdown function which shows the user the exact numbers of each repeat on that given chromosome which is linked to our MySQL database.

For this visualisation, the R libraries karyoploteR and BSgenome.Hapiens.UCSC.hg38 were used. The karyoploteR library was used to create ideograms for each of the chromosomes 1-22, X and Y. The HERV and LINE1 repeats were mapped to the ideogram according to their start and end positions on each chromosome. The BSgenome.Hapiens.UCSC.hg38 library was used to set the sequence lengths of the chosen chromosome to avoid incorrect mapping of the repeats. Two 'tracks' were visualised for each Superfamily, one for the mapping of the repeat regions to the appropriate position and the other for mapping the density of those repeats within a window of 100,000 base pairs. From this the user can easily see which part of the chromosome contains the most repeats for HERV and LINE1. This process created an image for each chromosome which could be exported from R.

To improve the ideogram, a track ruler was used to show the length in base pairs under each chromosome. For this a ‘tick’ distance of every 10 was set which represents 100,00 base pairs. The cytoband names for each chromosome was also added and adjusted to improve the visualisation.

The karyoploteR library was chosen as it allows users to easily customize how their data is visualised using a variety of different functions. It can visualise multiple data types and a lot of information in a single image which is important in a project such as this with a large amount of data.

Peptide sequence list

Relationship Viewer

This part of the website allows the user to click the 2 main buttons (HERV and LINE1) which displays their respective relationship tree image. The trees were created using a few different programs which allowed for the use of aligning the sequence, creating consensus sequences and finally displaying the relationship between the sequences. Muscle was used to create an alignment followed by Emboss Consensus to create a consensus sequence. Muscle was then used again to align all the sequences and a tree image file was created using an online resource tool called ITOL (Interactive tree of life).

Additional User Functionality – Upload User Tree

The software allows a user to upload a tree file of their own and the software will process the file and output a tree that they can visualise. The only accepted files currently are tree files in the newick format (.nwk, .newick) and .ph files.

Upon uploading a file, a new window displaying the tree file is shown. If the window does not appear, the website also automatically displays another version of the tree file in the website itself which is generated after submission of the file.

The Phylo module from Biopython is used to parse the file and then using matplotlib and phylo module, a tree image is displayed in a new pop up window which can be altered using limited functions such as zooming and panning around.

If an incorrect filetype is uploaded, the software rejects the file and displays an error message.

Uploading of Files

Allowing users to upload files of their own allows them to be able to search our database and identify family members that the peptide sequence has been translated from.

Peptide Sequence Identifier (Fasta format)

This webpage allows for user input via 2 different methods. The first method allows a user to upload a fasta sequence file to the server in order for it to analyse the file. The second method allows a user to paste a fasta sequence into a text box before pressing search.

The server first waits for a POST request method to engage before doing any further actions. If a GET request method is used, the normal webpage is served to the client with some data (up to 1000 rows) from the database being displayed. This allows the user to be able to navigate part of the database without any input.

Fasta Checker

Due to the possible memory constraints on the host machine, the option was taken to disallow entering in more than one fasta sequence and fasta sequences with headers via the text box. Instead the user is able to upload the file as a fasta file (including headers) where it can be parsed more efficiently using SeqIO module from biopython.

The text field allows the user to enter any length of peptide sequence with spaces and tabs in between and is able to concatenate and parse the data together (for ease of copy / paste use for the client).

If a file is uploaded that contains no data, an error message is relayed back to the user stating the file is empty.

Incorrect Filetypes

The program is able to differentiate between the correct and incorrect filetypes. If an incorrect filetype is uploaded, an error will be produced stating the user to upload a correct file with the correct extension.

Parsing correct files

If the correct requirements have been met, the peptide data is then queried using MYSQL and if a match has been found in the database, it returns the family name in a table.

Downloading results

Once data has been submitted and a result has been received, a table is displayed showing all the necessary information. The user also has the option to download the results in many different forms such as being able to copy, export to csv, export to excel, export to pdf and print the page. This is provided via 5 different buttons located above the table.

The programming

The website is able to differentiate between input via upload box and the search box and conducts different execution of the commands depending on the route the information has taken to arrive.

When a post method is received via the text box, the program does a fasta check by looking at the presence of headers and rejects the data if it has been detected. This decision was taken as parsing a file with headers would require more memory and time than would be preferred. The user is instead able to upload the file (with headers) for it to be parsed quicker.

If a post method is received via the upload function, the server moves to the correct file directory to where the file has been uploaded so that it can be read into memory. A few checks are done on the file to ensure that it is an appropriate filetype of the fasta format before being read in. Failure to meet the correct specifications of the file means that the website can provide different errors from incorrect file types to empty file found.

Once the checks have been passed, the Biopython module known as SeqIO is used to parse the file into memory and a further check is conducted to see the length of the sequences. If the length is equal to one, a simple MySQL query is conducted and returns the family name if a match has been found.

The program can also detect whether the total number of sequences is below or above 5000. If it is equal to or below, the program does a simple join of the sequences to create a long query and submits it to the MySQL server. If the number of sequences exceeds 5000, the program divides the sequences into chunks of 5000 and submits the query.

This was done to reduce the wait times and stress on the server. Recent optimisation of the code increased the efficiency by up to 10x.

Mzident / MzTab

This webpage allows a user to upload a file that can be either an mzident or mztab formatted file. A user can also select the tissue type and disease progression type before submitting the file if the information is known about where the data has been sourced from.

This part of the software is able to detect the different filetypes that are uploaded and will only accept files with the correct extension.

Parsing of files

The type of parsing that occurs is dependent on the type of file extension that is uploaded to the server. The server is able to analyse the extension and differentiate between the accepted and non-accepted forms and then use regular expressions to extract the information needed. The type of regex used differs between the type of file that has been uploaded. The extracted peptide sequence is then queried in the database via MYSQL and the resulting matches are displayed in a table.

Disease / Tissue Types

Before submitting a file, a user is able to select the tissue and disease type. This is then saved a lot with the family names that is retrieved from the MySQL query and is used for the expression atlas.

Hashing of files

The server conducts a hash check of the file to see if it has been uploaded before, if it has not been uploaded previously, the additional information is saved which contributes to the count tracker that is available on the expression atlas.

To ensure the integrity of the database and the validity of the data (specifically the number of counts / hits of the peptide sequences), files are hashed to create an ID which is then stored in a CSV file.

When a file is uploaded, a unique ID is created using SHA-224 cryptographic hash functions (from the module hashlib). This ID is then searched for in the CSV file and if no match is found, the ID is then appended to the end and the CSV file and a MySQL query is conducted saving the family name, tissue type and disease type that was selected by the user.

If the unique ID is found in the CSV, the data is not saved into MYSQL DB, however it is still analysed and provides the requested data back to the user.

Benchmarking Of Regex

Benchmarking was conducted on the regex function to ensure that this would not cause any problems when submitting large files (over 100mb) and satisfied the loading times that would be expected.

MZIDENT FILE	Sequences	Time (seconds)
12MB	2,458	1 – 1.50
59MB	44,780	3 – 3.50
111MB	22,122	5.50 – 6.50
1GB	??	??
MZTAB FILE		
16kb	36	<1
112MB	269,100	9.0 – 9.50
1GB	??	??

The results displayed show that the current code that was created to do the job seems adequate for the task required and that the amount of sequences plays a large role in the time taken compared to the size of the file.

Downloading results

Refer back to 14.1 which discusses this section.

Expression Atlas

This part of the application allows the user to view all the repeats which have been found by the peptide identification in uploaded files from previous users.

This works through interacting with the peptide identification part of the application which inserts information successfully identified from uploaded samples into the MySQL database into an expression atlas table. It also inserts the number of each tissue type and total number of samples added into the atlas into a separate table which is accessed when the user visits the expression atlas page on the website. When this occurs, the MySQL database is accessed through a query which returns the tissue type, number of repeats, each individual repeat found and the disease state from the uploaded samples. This data within the atlas can be viewed by the user in relation to the percentage of each tissue type compared to all other tissue types, or by the percentage of each family identified in a chosen tissue type. This was achieved through the use of multiple MySQL queries accessing different tables to calculate each percentage.

Limitations

During the process of this project, the idea of having an interactive visualisation of the distribution and relationship between the translated peptides did not become a reality. This was due to the time restraints and limited resources during the project.

$Fettucine Retrotransposon database project

Features:

- Family Table
- Distribution of retrotransposons
- Translated products
- Amino Acid relationship viewer
- Peptide Sequence identifier
- Mzid/mzTab peptide analyser
- Expression Atlas

Packages / Systems used:

- Flask
- Python
- MySQL
- JQuery / Javascript