**Help manual**

**Function**

This program is for retrieve a group of protein sequences, align them, determine the level of conservation, and scan the sequences of interest with motifs from the PROSITE database.

**Description**

1. Fetch protein sequences of a protein family in a specific taxonomic group from NCBI. Users can choose to fetch not partial sequences or all sequences.

2. Extract the information of the retrieved results and store it into a table. Calculate the number of species and genus it contains. Display the result customized by user. Ask the user to continue or not.

3. Sequence alignment. Sometimes the difference of sequences length in this group is too big: (max-min)/max > 0.2, the alignment will commonly give out bad result. Thus, the programme will display the distribution of sequence length based on statistic result and histogram, then ask the users if they want to choose an appropriate subgroup for alignment based on the range of sequence length.

4. Align the sequences of the subgroup, determine and plot the level of conservation, and generate consensus sequence. Ask user if they want to choose another subgroup according to the alignment result. If not, ask user if they want to continue finding motifs.

5. Generate fasta file of each sequence to scan each sequence with motifs from the PROSITE database. Then calculate the total number of each motif and display it.

**Applications or Modules that Needs to be Installed Ahead**

Applications: Python3, E-utilities, EMBOSS, clustalo

Python3 Models: os, pandas, re, matplotlib.pyplot, shutil

**Usage**

**For Ordinary User:**

Open a terminal window, set an empty working directory, type in:

|  |
| --- |
| mkdir <foldername> |
| cd <foldername> |

To get into the python environment, type in:

|  |
| --- |
| python3 |

To start the program, type in:

|  |
| --- |
| import ICA2\_function |
| exec(open('ICA2\_script.py').read()) |

While running the program, follow the instructions and type in the required input. Very easy.

**Warnings:**

Every time you want to re-run the program, you might consider restarting the terminal window to prevent any failure of generating or displaying plots. If the program fails when generating plot for no reason, please reopen the terminal window and run the program again.

**For Competent Python3 Code-writer:**

In the python3 environment, import " ICA2\_function" and execute "ICA2\_script.py" to start the program.

There are 7 functions in "ICA2\_function.py", and "ICA2\_script.py" is for calling and execute these functions.

**Function Details**

1. ICA2\_function.pro\_tx\_seq() let user input protein name and taxonomic group ID or name, ask user if they want to get only not partial sequences or all sequences. And then check if the input is valid or if the retrieve result is not 0. If the input is not valid or retrieve 0 result, let user input again until is valid. Then retrieve a query using “esearch” and get the counts of result. If the retrieve result is more than 1000 sequences, then ask the user if they want to download or not. If not, ask the user if they want to retype or just quit the program. Download the sequences in fasta format and save it into “<protaxpar\_label>.fa" file.

At the end, it returns a list [protein, taxonomy, protaxpar\_label] containing protein and taxonomy name and a label for naming the file. There are two kinds of "protaxpar\_label" based on the way of retrieve:

"<protein>\_<taxonomy>\_all" for retrieving all sequences and

"<protein>\_<taxonomy>\_no\_par" retrieving only not partial sequences.

|  |
| --- |
| dataset=ICA2\_function.pro\_tx\_seq() |

2. ICA2\_function.create\_df(protaxpar\_label) function is for making a dataframe with 9 columns: ['protein\_accession', 'protein\_name', 'genus', 'species', 'full\_species\_name', 'length', 'header', 'sequence', 'raw\_seq’]. It splits the header go get genus and species of each protein squence, calculates the sequence length. (For some partial sequences or other kinds of sequences, it sometimes has unclear header(don’t have “[species]” format), then the columns extracted from header will simply equals to all contents after accession number. At the end, return the dataframe and output it into “<protaxpar\_label>\_summary.csv” file.

|  |
| --- |
| df=ICA2\_function.create\_df(dataset[2]) |

3. ICA2\_function.count\_species(protaxpar\_label) function can check the diversity of species and genus of the result. Load the dataframe and count the number of unique genus and species and total number of proteins. The percentage of protein per genus or protein per species is also calculated. Print those results, display the first 10 rows of the dataframe with columns [‘protein\_accession',"protein\_name",'genus','species'] sorted by genus name and save the dataframe into “<protaxpar\_label>\_summary\_species.csv” file. User can choose to see number of head rows or tail rows or choose to see the rows of specific genus multiple times in a loop.

|  |
| --- |
| ICA2\_function.count\_species(protaxpar\_label) |

Use a conditional check ask if the user want to continue for the alignment. If not, quit the program.

4. ICA2\_function.seq\_len\_subset(dataset) function checks the distribution of sequence length and generate a subset group if needed. First, print out the calculation summary of sequence length, display the histogram of the distribution of sequence length, save the histogram into "<protaxpar\_label>.seqLenHis.png". If the difference of sequences length is too big: (maxLen-minLen)/maxLen > 0.2, then ask the user if they want to choose a subset for the following analysis based on the range of sequence length. User input the minimum and maximum length, run in loop until the input is valid. Generate a new dataframe that contain only the sequences with length inside that range. Output the filtered dataframe into "<sub\_label>\_subgroup.csv" file and the filtered sequences into "<sub\_label>\_len.fa" file. Return the filename label of the subset.

There are two kinds of "sub\_label" based on the length range.

If user chose a minimum and maximum length:

sub\_label = <protaxpar\_label>\_<min>\_<max>

If not choosing a length range:

sub\_label = <protaxpar\_label>\_whole

|  |
| --- |
| sub\_label=ICA2\_function.seq\_len\_subset(dataset) |

5. ICA2\_function.conservation(dataset,sub\_label) is to determine, and plot, the level of protein sequence conservation. Align the sequences using "clustalo", output to "<sub\_label>.align.msf" file. Get basic information for the alignment using "infoalign" function. Make the information into dataframe, sort it by "Identity" from high to low and print on screen, output to "<sub\_label>.infoalign.csv" file. Generate conservation sequence using "cons" function, save it into "<sub\_label>\_align.cons" file and display it on screen. Plot the level of conservation usning "plotcon" function and save into "<sub\_label>\_plotcon.png " file, display on screen.

|  |
| --- |
| ICA2\_function.conservation(dataset,sub\_label) |

Ask user if they want to choose a new subset and redo the alignment, or continue scanning with motif, or quit the program.

6. ICA2\_function.seq\_out(sub\_label) function is to output fasta file for each sequence in the subgroup.

|  |
| --- |
| ICA2\_function.seq\_out(sub\_label) |

7. ICA2\_function.find\_motifs(sub\_label) is to scan with motifs from PROSITE database. First, scan motif for each sequence in parallel, get all the motif names from all files, count the number of each motif in total. Display the counts of motifs sorted in a descending order. Save the result into "<sub\_label>patmatmotifs\_summary.csv" file.

|  |
| --- |
| ICA2\_function.find\_motifs(sub\_label) |

Diagram, schematic

Description automatically generated**FlowChart:**

Diagram, schematic

Description automatically generated

**Output Example:**

Use pyruvate dehydrogenase from ascomycete fungi (taxonID 4890) as example:

>>> import ICA2\_function

>>> exec(open('ICA2\_script.py').read())

Please type in protein name:

pyruvate dehydrogenase

Please type in a taxonomic name or a taxonomy ID. (taxonomy ID for example: txid4890)

ascomycete fungi

Generate query for "pyruvate dehydrogenase" protein and "ascomycete fungi" taxonomic group.

would you like to contain partial proteins or not? yes/no:

no

searching no partial protein sequences from ncbi...

208 sequences downloading...

protein sequences is inside file: pyruvate\_dehydrogenase\_ascomycete\_fungi\_.fa

…

Only showing the first few lines.

**Output file:**

|  |  |
| --- | --- |
| <protaxpar\_label>.fa | retrieved sequences |
| <protaxpar\_label>\_summary.csv | sequences information table |
| <protaxpar\_label>.seqLenHis.png | histogram of sequences length |
| <protaxpar\_label>\_summary\_species.csv | protein and corresponding species |
| <sub\_label>\_subgroup.csv | subgroup information table |
| <sub\_label>\_len.fa | sequences of subgroup |
| <sub\_label>.align.msf | aligned sequences |
| <sub\_label>.infoalign.csv | alignment result of each sequence |
| <sub\_label>\_align.cons | conservation sequence |
| <sub\_label>\_plotcon.png | Image showing the level of conservation |
| <sub\_label>patmatmotifs\_summary.csv | result summary of scanning with motifs |

In the "<sub\_label>.infoalign.csv " file, the description of each column are shown here.

* Name - name of the sequence.
* SeqLen - length of the sequence when all gap characters are removed.
* AlignLen - length of the sequence including internal gap characters i.e. gaps at the start or the end are not included.
* Gaps - number of gaps e.g. 'AAA---AAA' is 1 gap (and 3 gap characters long (see GapLen)).
* GapLen - total number of internal gap characters, see the 3 gap characters above. This is the sum total of all of the internal gap characters in this sequence.
* Ident - number of characters that are identical to the specified reference sequence (uppercase 'A' is identical to lowercase 'a').
* Similar - number of characters which are non-identical - which score > 0 in the comparison matrix when compared to the reference sequence, but which are not identical.
* Different - number of characters which score <= 0 in the comparison matrix when compared to the reference sequence.
* %Change - a simple measure of the percentage change as compared to the reference sequence: (AlignLen - Ident) \* 100 / AlignLen