Protein Analysis.py

A protein alignment and analysis script

Table of Contents

Zipfile Name: Ish-2019.Assignment2.zip

Zipfile location: https://github.com/IshIshIsh/Learning/tree/master/BPSM_assignment2

Zipfile Password: Python

Script Overview

PURPOSE

The protein analysis pipeline is a python script which allows the user to download and analyze a set of proteins from a single or group of specified gene(s) and taxonomic group(s). To allow greater flexibility, sub-processes are grouped by task and can run independently if the directory and files are specified correctly. An alternative approach, is to run the master function: master_analysis(...) which will automatically parse the outputs of each sub-process into the next sub-process in one batch.

Analysis overview with group main function (see relevant section for more details).

- Sequence Retrieval
  - Obtain relevant protein sequence data from NCBI
  - Function: master_seq_retreive(...)
- Sequence Alignment:
  - Generate an alignment of selected proteins and create a consensus sequence
  - Function: master_clustalo(...)
- Protein statistics:
  - Generates protein statistics per protein sequence.
  - Function: master_protein_analysis(...)
- Blastp:
  - Analyses how similar a query sequence is to selected proteins. Using a consensus sequence as a query will provide a general measure of relatedness of all sequences.
  - Function: master_blast(...)
- Motif searching:
  - Analyze what motifs are present in a set of protein sequences.
  - Function: master_motifs(...)
- Protein & Conservation plots
  - Create plots of conservation between alignments, hydropathy across all input sequences and single-protein summary plots.
  - Function: master_graphs(...)

MASTER FUNCTION

INPUTS

The script requires only one or more taxon id(s) and gene name(s) although by default the user is prompted for several basic decisions (described in appropriate sections) with information provided on the input type expected and the default values. The user can choose whether to overwrite these values with a custom entry or use the defaults by pressing enter. An option has been provided if a user wishes to only use the default values; passing in defaults_requests = True will prevent any user input dialog. A more advanced option is also provided, using the **kwargs argument to pass in several key word arguments, bypassing the user input requests.

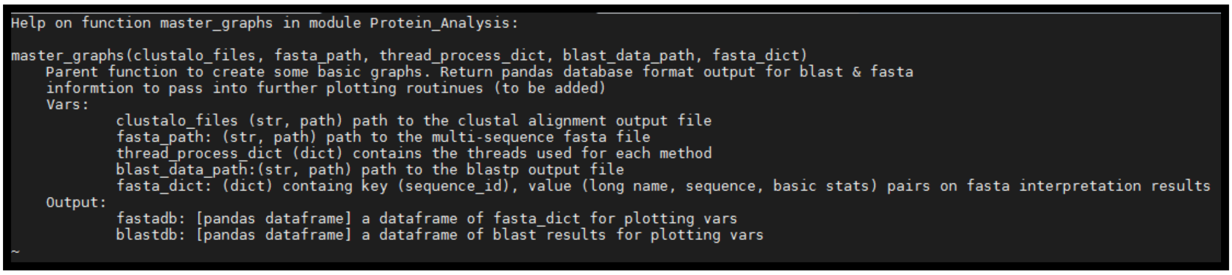
INPUT PARAMETERS AND DESCRIPTION

PARAMETER	DEFAULT	DESCRIPTION
Gene_family	NONE	A single gene name eg ‘COX1’ or a list of gene names in the format: “[‘G6pc’,’COX1’]”
Taxid	NONE	A single taxonomic id eg ‘txid8782’ or a list of taxonomic ids in the format: “[‘txid8782’,’txid40674’]”
Filtered_predicted **Filtered_partial**	True	Used in the sequence retrieval method. If these are true will ignore any files within the search term tagged as predicted or partial respectively. To include these sequences, set filtered_predicted and/or filtered_partial to False
Name	‘’	If no name is provided, the script will automatically create one from the gene name(s) and taxonomic id(s) provided. The name is used to create a folder to store the analysis (with the prefix ‘Protein Analysis’ and date suffix) and to label some output files.
Folder_time	False	If set to True, the analysis folder name will have a suffix of the hour_minutes when the analysis was run. Setting this variable to True allows you to compare analysis run on the same dataset on the same day.
Working_directory	String	If this is not specified the script will create the analysis folder within the users home directory. To change the directory of the output pass in a valid path extension.
Save_summary	True	If this is set to True, general summary reports which are printed to the screen will also be stored as txt files.
Keep_fastas	False	If this is set to True, individual fastas will be created for sequences selected for motif searching. Warning: If analyzing large datasets this may create space issues.
Silent	True	If this is set to False, the output of many functions will be printed to screen
Bin_no	250	The maximum number of sequences in each bin (a group of sequences based on their hierarchy in the blast search output)

Outputs

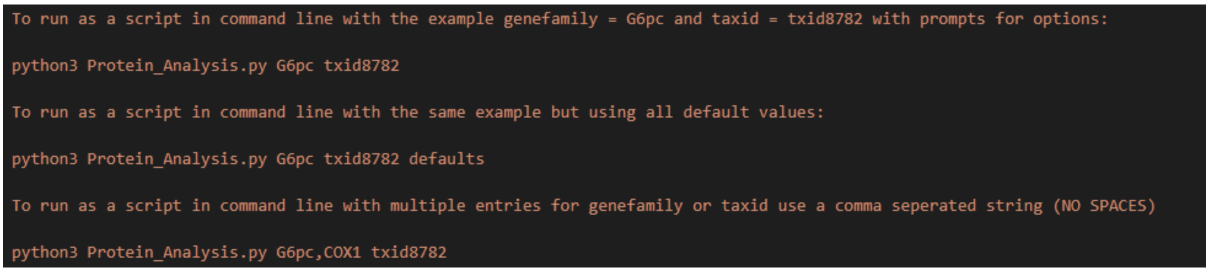
A main working directory is automatically generated to store the outputs of the script. By default, all processes will display a short output message on screen pointing the user to the location of the output files. Some processes will also generate and display a short summary file to give the user the required information to select inputs or continue to the next process. To get this data in long form printed to screen set the variable silent = False in the input line. All outputs are stored in the main working directory except the motif results from the blastp bin results which are stored in working_directory/motifs/bin_number/sequence_id.

Each function within the script contains a docstring pointing the user to helpful information on the inputs required and outputs produced which can be accessed by typing help(function_name).



```
Help on function master_graphs in module Protein_Analysis:

master_graphs(clustalo_files, fasta_path, thread_process_dict, blast_data_path, fasta_dict)
    Parent function to create some basic graphs. Return pandas database format output for blast & fasta
    informtion to pass into further plotting routinues (to be added)
    Vars:
            clustalo_files (str, path) path to the clustal alignment output file
            fasta_path: (str, path) path to the multi-sequence fasta file
            thread_process_dict (dict) contains the threads used for each method
            blast_data_path:(str, path) path to the blastp output file
            fasta_dict: (dict) containg key (sequence_id), value (long name, sequence, basic stats) pairs on fasta interpretation results
    Output:
            fastadb: [pandas dataframe] a dataframe of fasta_dict for plotting vars
            blastdb: [pandas dataframe] a dataframe of blast results for plotting vars
~
```

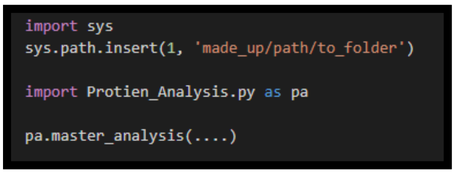
Use in Unix:



```
To run as a script in command line with the example genefamily = G6pc and taxid = txid8782 with prompts for options:

python3 Protein_Analysis.py G6pc txid8782

To run as a script in command line with the same example but using all default values:

python3 Protein_Analysis.py G6pc txid8782 defaults

To run as a script in command line with multiple entries for genefamily or taxid use a comma seperated string (NO SPACES)

python3 Protein_Analysis.py G6pc,COX1 txid8782
```

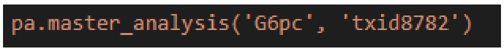
Use in Python:

To use the program through import please store the Protein_Analysis.py file and the __init__.py file within your working directory. Alternatively import the sys module and add sys.path.insert(1, 'path/to/folder') eg:



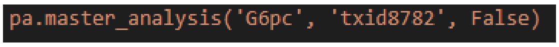
```
import sys
sys.path.insert(1, 'made_up/path/to_folder')

import Protien_Analysis.py as pa

pa.master_analysis(....)
```

To run from a Python3 shell with user prompts for parameter values:



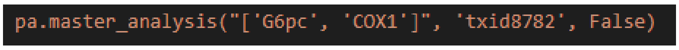
```
pa.master_analysis('G6pc', 'txid8782')
```

To run from a Python3 shell using only default values:



```
pa.master_analysis('G6pc', 'txid8782', False)
```

To run from a Python3 shell with multiple gene names:



```
pa.master_analysis("['G6pc', 'COX1']", 'txid8782', False)
```

Sequence Retrieval

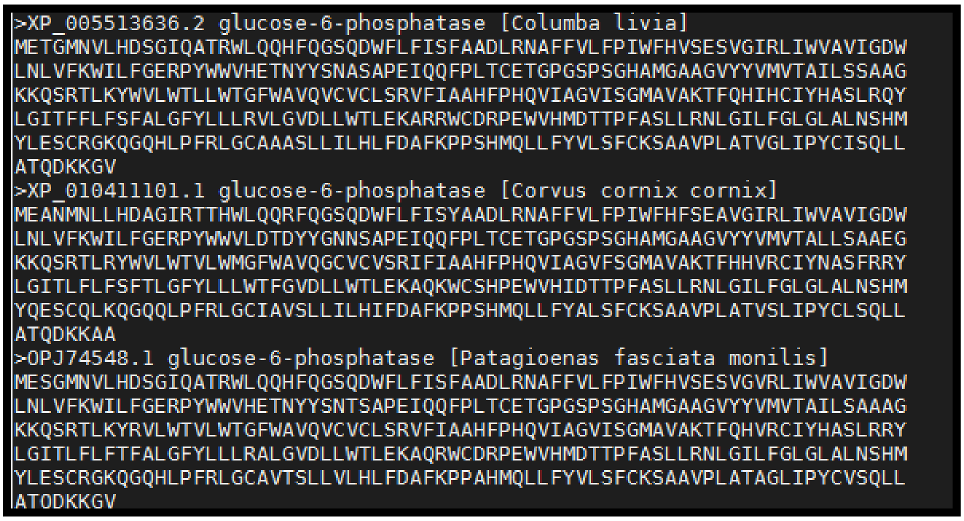
Sequence retrieval can be carried out independently using the master_seq_retrieve function

Purpose & Use

The sequence retrieval method uses the input gene family(s) and taxid(s) to query the NCBI protein database and generate a single .fasta file containing multiple sequences with identifying information saved to a file with the suffix _seqs.fasta. The output from this method is stored in a folder either named by the users or automatically generated using the gene_family_taxid and date of analysis. A summary report is generated on screen and stored in NCBI_summary.txt.

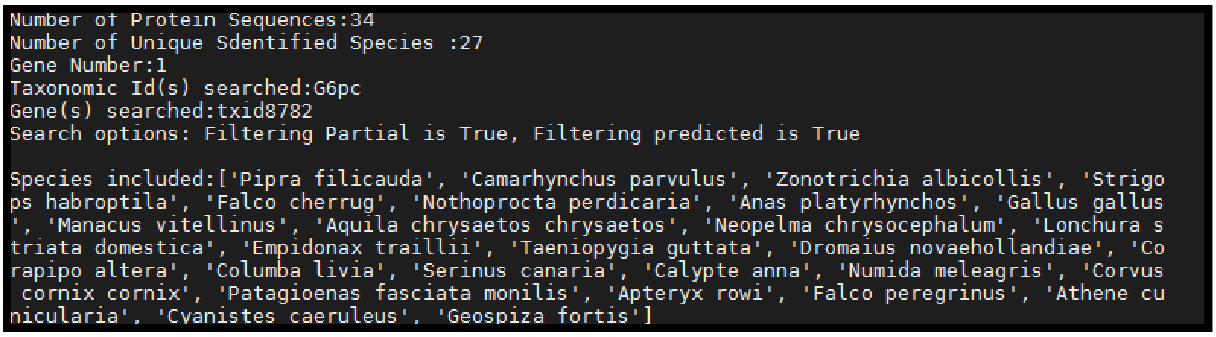
Outputs

Standard Output for [name]_seqs.fasta



```
>XP_005513636.2 glucose-6-phosphatase [Columba livia]
METGMNVLHDSGIQATRWLQQHFQGSQDWFLFISFAADLRNAFFVLFPIWFHVSESVGIRLIWVAVIGDW
LNLVFKWILFGERPYWWVHETNYYSNASAPEIQQFPLTCETGPGSPSGHAMGAAGVYYVMVTAILSSAAG
KKQSRTLKYWVLWTLLWTGFWAVQVCVCLSRVFIAAHFPHQVIAGVISGMAVAKTFQHIHCIYHASLRQY
LGITFFLFSFALGFYLLLRVLGVDLLWTLEKARRWCDRPEWVHMDTTPFASLLRNLGILFGLGLALNSHM
YLESCRGKQGQHLPFRLGCAAASLLILHLFDAFKPPSHMQLLFYVLSFCKSAAVPLATVGLIPYCISQLL
ATQDKKGV
>XP_010411101.1 glucose-6-phosphatase [Corvus cornix cornix]
MEANMNLLHDAGIRTTHWLQQRFQGSQDWFLFISYAADLRNAFFVLFPIWFHFSEAVGIRLIWVAVIGDW
LNLVFKWILFGERPYWWVLDTDYYGNNSAPEIQQFPLTCETGPGSPSGHAMGAAGVYYVMVTALLSAAEG
KKQSRTLRYWVLWTVLWMGFWAVQGCVCVSRIFIAAHFPHQVIAGVFSGMAVAKTFHHVRCIYNASFRRY
LGITLFLFSFTLGFYLLLWTFGVDLLWTLEKAQKWCSHPEWVHIDTTPFASLLRNLGILFGLGLALNSHM
YQESCQLKQGQQLPFRLGCIAVSLLILHIFDAFKPPSHMQLLFYALSFCKSAAVPLATVSLIPYCLSQLL
ATQDKKAA
>OPJ74548.1 glucose-6-phosphatase [Patagioenas fasciata monilis]
MESGMNVLHDSGIQATRWLQQHFQGSQDWFLFISFAADLRNAFFVLFPIWFHVSESVGVRLIWVAVIGDW
LNLVFKWILFGERPYWWVHETNYYSNTSAPEIQQFPLTCETGPGSPSGHAMGAAGVYYVMVTAILSAAAG
KKQSRTLKYRVLWTVLWTGFWAVQVCVCLSRVFIAAHFPHQVIAGVISGMAVAKTFQHVRCIYHASLRRY
LGITLFLFTFALGFYLLLRALGVDLLWTLEKAQRWCDRPEWVHMDTTPFASLLRNLGILFGLGLALNSHM
YLESCRGKQGQHLPFRLGCAVTSLLVLHLFDAFKPPAHMQLLFYVLSFCKSAAVPLATAGLIPYCVSQLL
ATQDKKGV
```

Standard Output for NCBI_summary.txt



```
Number of Protein Sequences:34
Number of Unique Sdentified Species :27
Gene Number:1
Taxonomic Id(s) searched:G6pc
Gene(s) searched:txid8782
Search options: Filtering Partial is True, Filtering predicted is True

Species included:['Pipra filicauda', 'Camarhynchus parvulus', 'Zonotrichia albicollis', 'Strigo
ps habroptila', 'Falco cherrug', 'Nothoprocta perdicaria', 'Anas platyrhynchos', 'Gallus gallus
', 'Manacus vitellinus', 'Aquila chrysaetos chrysaetos', 'Neopelma chrysocephalum', 'Lonchura s
triata domestica', 'Empidonax traillii', 'Taeniopygia guttata', 'Dromaius novaehollandiae', 'Co
rapipo altera', 'Columba livia', 'Serinus canaria', 'Calypte anna', 'Numida meleagris', 'Corvus
 cornix cornix', 'Patagioenas fasciata monilis', 'Apteryx rowi', 'Falco peregrinus', 'Athene cu
nicularia', 'Cyanistes caeruleus', 'Geospiza fortis']
```

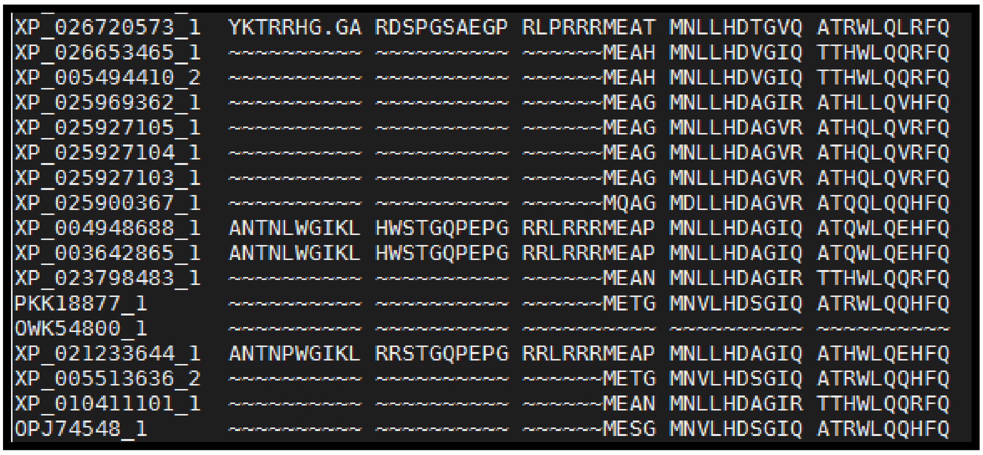
Clustalo Sequence Alignment

Clustalo alignment can be carried out independently using the master_clustalo function

Alignment:

The script will generate a clustalo alignment of all sequences in the fasta file returned from the NCBI sequence retrieval script and save the output as [name]_clustal.msf.

Example of clustal output format:

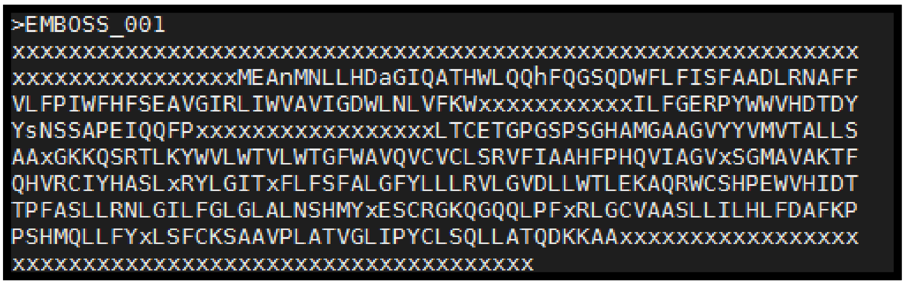


```
XP_026720573_1  YKTRRHG.GA RDSPGSAEGP RLPRRRMEAT MNLLHDTGVQ ATRWLQLRFQ
XP_026653465_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAH MNLLHDVGIQ TTHWLQQRFQ
XP_005494410_2  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAH MNLLHDVGIQ TTHWLQQRFQ
XP_025969362_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAG MNLLHDAGIR ATHLLQVHFQ
XP_025927105_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAG MNLLHDAGVR ATHQLQVRFQ
XP_025927104_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAG MNLLHDAGVR ATHQLQVRFQ
XP_025927103_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAG MNLLHDAGVR ATHQLQVRFQ
XP_025900367_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MQAG MDLLHDAGVR ATQQLQQHFQ
XP_004948688_1  ANTNLWGIKL HWSTGQPEPG RRLRRRMEAP MNLLHDAGIQ ATQWLQEHFQ
XP_003642865_1  ANTNLWGIKL HWSTGQPEPG RRLRRRMEAP MNLLHDAGIQ ATQWLQEHFQ
XP_023798483_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAN MNLLHDAGIR TTHWLQQRFQ
PKK18877_1      ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~METG MNVLHDSGIQ ATRWLQQHFQ
OWK54800_1      ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~
XP_021233644_1  ANTNPWGIKL RRSTGQPEPG RRLRRRMEAP MNLLHDAGIQ ATHWLQEHFQ
XP_005513636_2  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~METG MNVLHDSGIQ ATRWLQQHFQ
XP_010411101_1  ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MEAN MNLLHDAGIR TTHWLQQRFQ
OPJ74548_1      ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~MESG MNVLHDSGIQ ATRWLQQHFQ
```

Consensus Creation

By default, the script will create a consensus file from the sequence alignment, print the consensus sequence to screen and save the output as [name]_consensus.fa.

Example of consensus.fa format:



```
>EMBOSS_001
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxMEAnMNLLHDaGIQATHWLQQhFQGSQDWFLFISFAADLRNAFF
VLFPIWFHFSEAVGIRLIWVAVIGDWLNLVFKWxxxxxxxxxxxILFGERPYWWVHDTDY
YsNSSAPEIQQFPxxxxxxxxxxxxxxxxxLTCETGPGSPSGHAMGAAGVYYVMVTALLS
AAxGKKQSRTLKYWVLWTVLWTGFWAVQVCVCLSRVFIAAHFPHQVIAGVxSGMAVAKTF
QHVRCIYHASLxRYLGITxFLFSFALGFYLLLRVLGVDLLWTLEKAQRWCSHPEWVHIDT
TPFASLLRNLGILFGLGLALNSHMYxESCRGKQGQQLPFxRLGCVAASLLILHLFDAFKP
PSHMQLLFYxLSFCKSAAVPLATVGLIPYCLSQLLATQDKKAAxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

BLASTp Sequence Analysis

Blastp can be carried out independently using the master_blast function

Create blastdb

Blastp is a method of checking the relatedness of a query sequence to a database of related sequences. The blastdb database can be downloaded from NCBI as a general database or a custom database can be made using blastdb from the multi-sequence fasta file.

Query choice

To enable the greatest flexibility in searching, any query sequence present in the Sequences folder can be used to blast against the database. The script checks all sequences with file extension .fa or .fasta and prompts the user to select one. The consensus sequence created from the clustalo alignment to generate a ranking of how related each sequence in the database is to each other. If a multi-sequence fasta file is selected, the user is prompted to select a single sequence_id from a list of the sequences in the fasta file.

Blastp Output

The output file generated uses the query_sequence_id as a prefix to _blastp_output.txt so multiple query sequences can be analysed consecutively all outputs stored within the same "Sequences" folder. The output is a ranked table tab separated text file:

```
BLASTP 2.5.0+
# Query: EMBOSS_001
# Database:
/localdisk/home/ifarquha/Protein_Analysis_G6pc_txid8782_2019_11_12/G6pc_txid8782_blastdb
# Fields: query acc., subject acc., % identity, alignment length, mismatches, gap opens, q. start, q. end, s.
start, s. end, evalue, bit score
# 34 hits found
EMBOSS_001    XP_029877954.1  86.010  386   25   3   77   462   1   357   0.0 666
EMBOSS_001    XP_027751872.1  85.530  387   27   3   77   463   1   358   0.0 662
EMBOSS_001    XP_030364364.1  85.013  387   29   3   77   463   1   358   0.0 660
EMBOSS_001    XP_027520533.1  85.271  387   28   3   77   463   1   358   0.0 659
EMBOSS_001    XP_027601652.1  85.271  387   28   3   77   463   1   358   0.0 659
EMBOSS_001    XP_005513636.2  84.715  386   30   3   77   462   1   357   0.0 656
EMBOSS_001    PKK18877.1    84.715  386   30   3   77   462   1   357   0.0 656
EMBOSS_001    XP_005439497.2  85.233  386   28   3   77   462   1   357   0.0 6
```

Result bins

The results from the blastp analysis are used to bin the input fasta sequences in order of most similar to least similar, with a default bin value of 250. These bins can then be used for further analysis such as protein information visualisation or motif searching. By default bin_0 is selected for further analysis; however a user prompt allows the selection of a specific bin or a list of bins for further analysis.

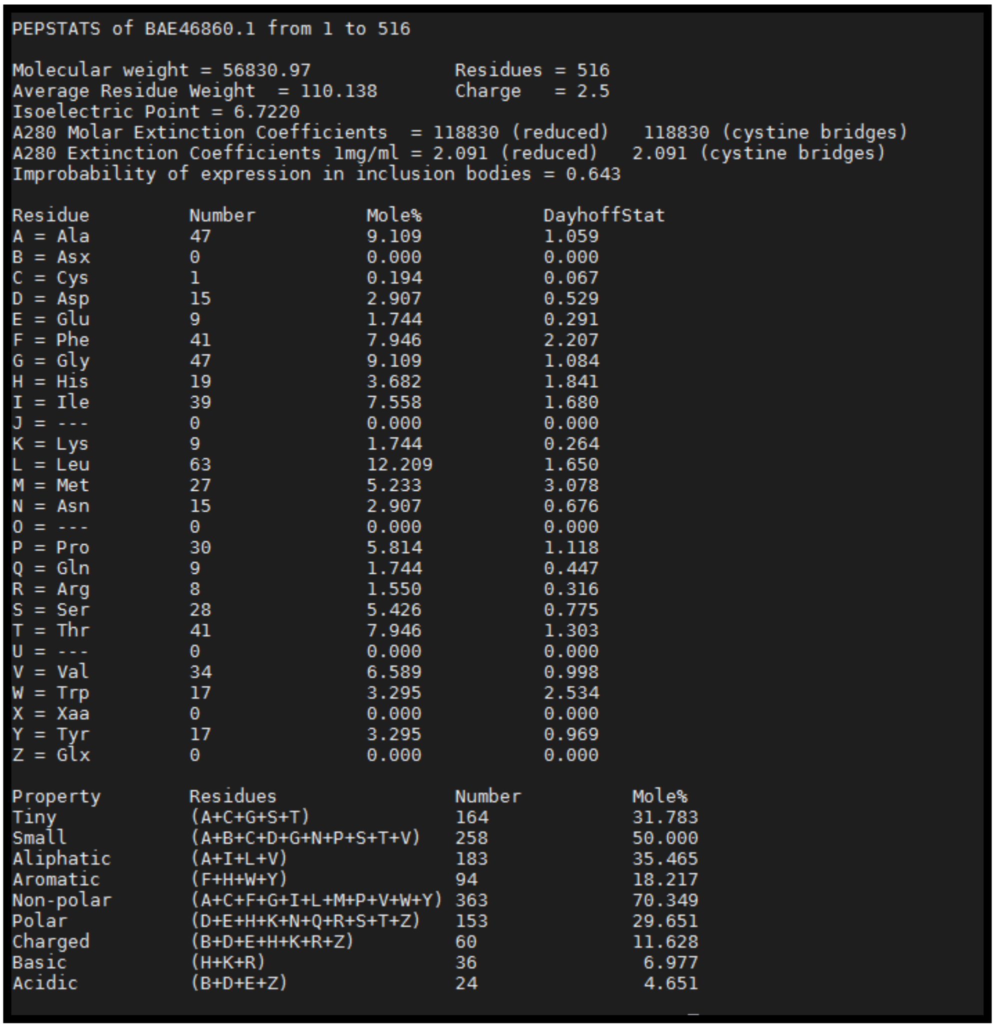
Protein Statistics & Fasta_dict

Protein analysis can be carried out independently using the master_protein_analysis function. Note the output fasta_dict is required for subsequent task-group functions.

PEPSTATS

The input fastas can be analysed with PEPSTATS to produce a report of protein statistics for an input fasta containing one or many sequences. The output of this report is stored in the file: [name]_stats.pepstats. Using this information and the information contained in the multi-sequence fasta file, a dictionary of results named fasta_dict is created and passed into subsequent functions.

Example output of one fasta records analysis from pepstats results file:



```
PEPSTATS of BAE46860.1 from 1 to 516

Molecular weight = 56830.97           Residues = 516
Average Residue Weight  = 110.138     Charge   = 2.5
Isoelectric Point = 6.7220
A280 Molar Extinction Coefficients  = 118830 (reduced)   118830 (cystine bridges)
A280 Extinction Coefficients 1mg/ml = 2.091 (reduced)   2.091 (cystine bridges)
Improbability of expression in inclusion bodies = 0.643

Residue         Number          Mole%           DayhoffStat
A = Ala         47              9.109           1.059
B = Asx         0               0.000           0.000
C = Cys         1               0.194           0.067
D = Asp         15              2.907           0.529
E = Glu         9               1.744           0.291
F = Phe         41              7.946           2.207
G = Gly         47              9.109           1.084
H = His         19              3.682           1.841
I = Ile         39              7.558           1.680
J = ---         0               0.000           0.000
K = Lys         9               1.744           0.264
L = Leu         63              12.209          1.650
M = Met         27              5.233           3.078
N = Asn         15              2.907           0.676
O = ---         0               0.000           0.000
P = Pro         30              5.814           1.118
Q = Gln         9               1.744           0.447
R = Arg         8               1.550           0.316
S = Ser         28              5.426           0.775
T = Thr         41              7.946           1.303
U = ---         0               0.000           0.000
V = Val         34              6.589           0.998
W = Trp         17              3.295           2.534
X = Xaa         0               0.000           0.000
Y = Tyr         17              3.295           0.969
Z = Glx         0               0.000           0.000

Property        Residues                Number          Mole%
Tiny            (A+C+G+S+T)             164             31.783
Small           (A+B+C+D+G+N+P+S+T+V)   258             50.000
Aliphatic       (A+I+L+V)               183             35.465
Aromatic        (F+H+W+Y)               94              18.217
Non-polar       (A+C+F+G+I+L+M+P+V+W+Y) 363             70.349
Polar           (D+E+H+K+N+Q+R+S+T+Z)   153             29.651
Charged         (B+D+E+H+K+R+Z)         60              11.628
Basic           (H+K+R)                 36               6.977
Acidic          (B+D+E+Z)               24               4.651
```

Motif Analysis

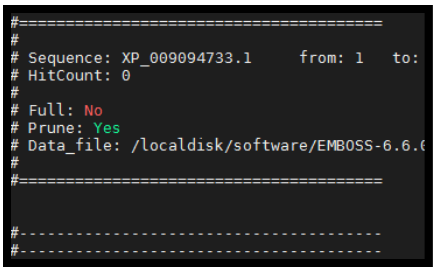
Motif analysis can be carried out independently using the master_motifs function

Prosite with Prosextract

The motif searching is performed using a database called PROSITE, which contains motifs (biologically relevant sequence patterns). To use the PROSITE search method certain files must be downloaded from the EBI database and constructed into a searchable database through PROSEXTRACT. The protein analysis script will check if the PROSITE database is present in an accessible format, and if not will download and compile the database automatically.

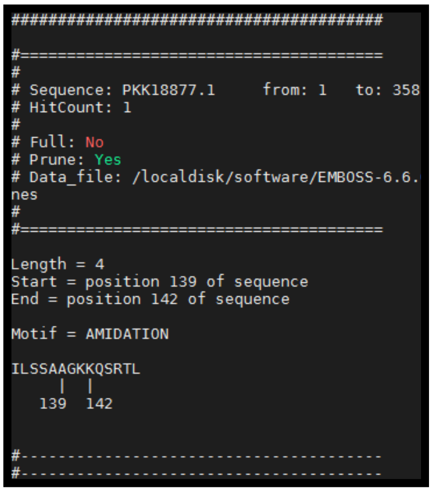
Patmatmotifs

If PROSITE database is present or after it has been compiled, the script will run the PROSITE motif search using the PATMATMOTIFS. This uses an individual fasta file (one per sequence) and searches for any known motifs. The output is then stored in a [sequence_id].motif file within the bin folder. Motif outputs will be in the following format if no motifs were found in the sequence:



```
#=======================================
#
# Sequence: XP_009094733.1     from: 1   to:
# HitCount: 0
#
# Full: No
# Prune: Yes
# Data_file: /localdisk/software/EMBOSS-6.6.(
#
#=======================================


#---------------------------------------
#---------------------------------------
```

If motif(s) were found in the given sequence, the output format will be:



```
########################################

#=======================================
#
# Sequence: PKK18877.1     from: 1   to: 358
# HitCount: 1
#
# Full: No
# Prune: Yes
# Data_file: /localdisk/software/EMBOSS-6.6.
nes
#
#=======================================

Length = 4
Start = position 139 of sequence
End = position 142 of sequence

Motif = AMIDATION

ILSSAAGKKQSRTL
     |  |
   139  142


#---------------------------------------
#---------------------------------------
```

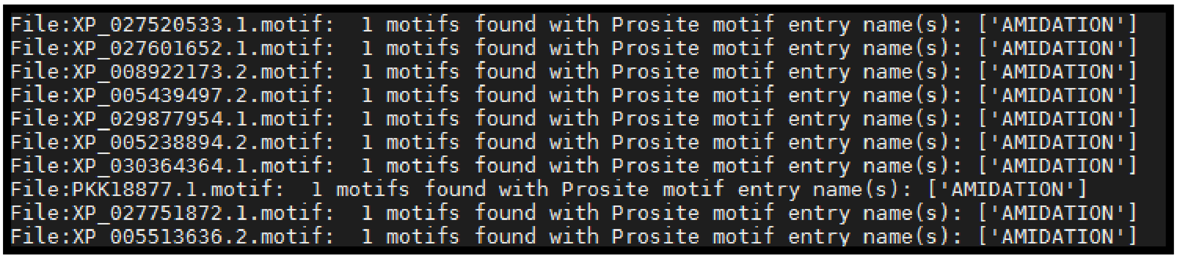
Motif reporting

A summary of the motifs found per bin is printed to screen and (optionally) saved to a txt file of motif_summary.txt in the relevant bin folder. This allows the user to easily find which motifs are present or common in each bin of sequences and enables searching of the PROSITE database using the motif pattern name to identify more information about the motifs found. NOTE: You can extract save the motif searching output in a long-form, which details all information on each motif by changing the default value output_format to 'long'. This will increase the file size output but reduces steps required to get all relevant information.

Format of motif summary:

Sequence_ID_A has 3 motif(s) found of motif pattern(s): [motif1, motif2, motif3]
Sequence_ID_B has 1 motif(s) found of motif pattern(s): [motif1]

Example output for motif summary:

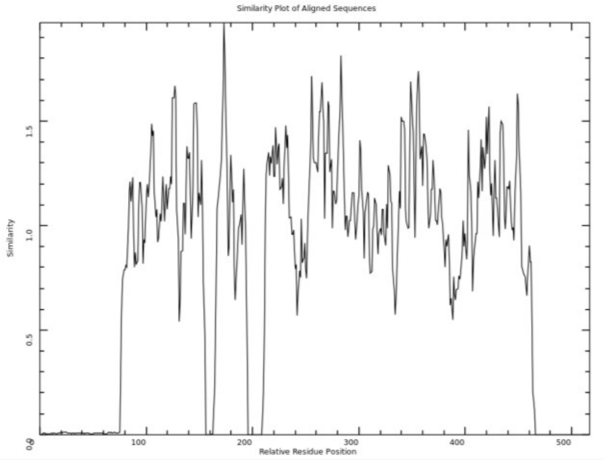


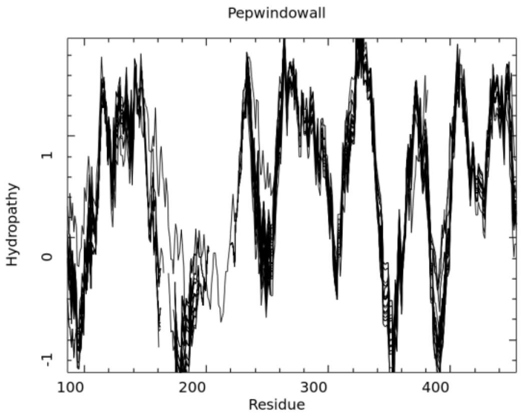
```
File:XP_027520533.1.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_027601652.1.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_008922173.2.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_005439497.2.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_029877954.1.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_005238894.2.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_030364364.1.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:PKK18877.1.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_027751872.1.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
File:XP_005513636.2.motif:  1 motifs found with Prosite motif entry name(s): ['AMIDATION']
```

Graphing

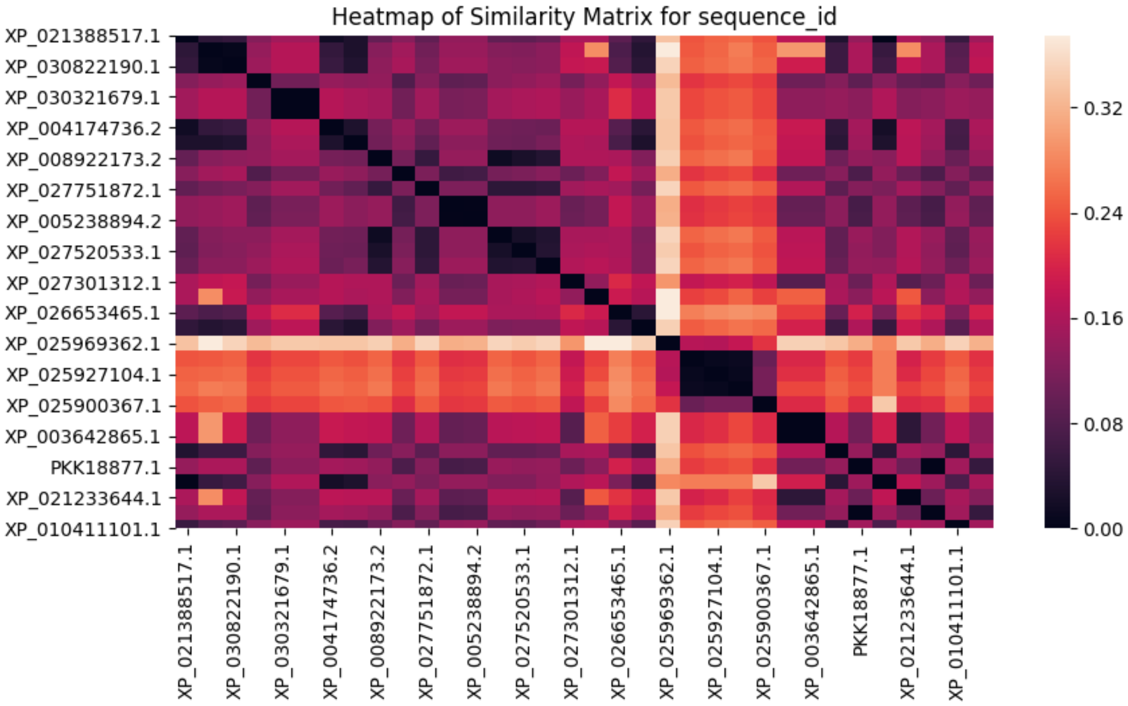
The master_graphs function task will output a number of plots which may be helpful in the analysis of your selected protein sequences. PLOTCON is used to create a similarity plot from the clustalo alignment file, which shows which regions were more conserved between the analysed proteins. Sequence similarity and conservation can also be assessed through the heatmap file from the similarity matrix produced with clustalo. A similarity matrix can be calculated using CLUSTALO and saved as file [name]_matrix.txt. This process can take a long time if many sequences are given, so the user is prompted whether they wish to continue with this part of the analysis. The similarity matrix is then parsed into a dataframe format, using the sequence_id as both index and column headers and plotted using the Seaborn heatmap package.

Standard output for Similarity Plot (plotcon) and hydropathy (pepwindowall)





Standard output for heatmap of pairwise distance matrix:



Maintaince Manual

Directories & files

The script automatically produces a directory to store the output, by default in the user's home space or within a specified folder (if passed in as a variable). The name of this level 0 working directory can be customised using input parameters within the master function. If this parameter is left as default, the directory name will be created using a prefix ("Protein Analysis") the taxid and family name passed into the function as parameters with a suffix of the date the script was run. An optional Boolean parameter of 'time' can be entered which adds the hours and minutes to the date used for automatic name generation allowing the user to compare the outputs of the same family/taxid combination with different evaluation parameters within the same day.

Within the level 0 working directory, separate directories will be created for:

1. Sequences and sequence information
2. Protein information
3. Motif information
   a. Bin result folder 1
   b. Bin result folder 2

The files produced by the script and any sub-folders (excluding level 0 workspace) are automatically named based on the taxid, family group and the output type. To ensure the user can locate all files, a message displayed on screen shows the full path extension to the relevant files when output is generated.

Warning: Renaming any of these files while the script is processing will result in errors. Please ensure any renaming or moving of files is carried out only when the full process has completed.

Table of Folder Outputs:

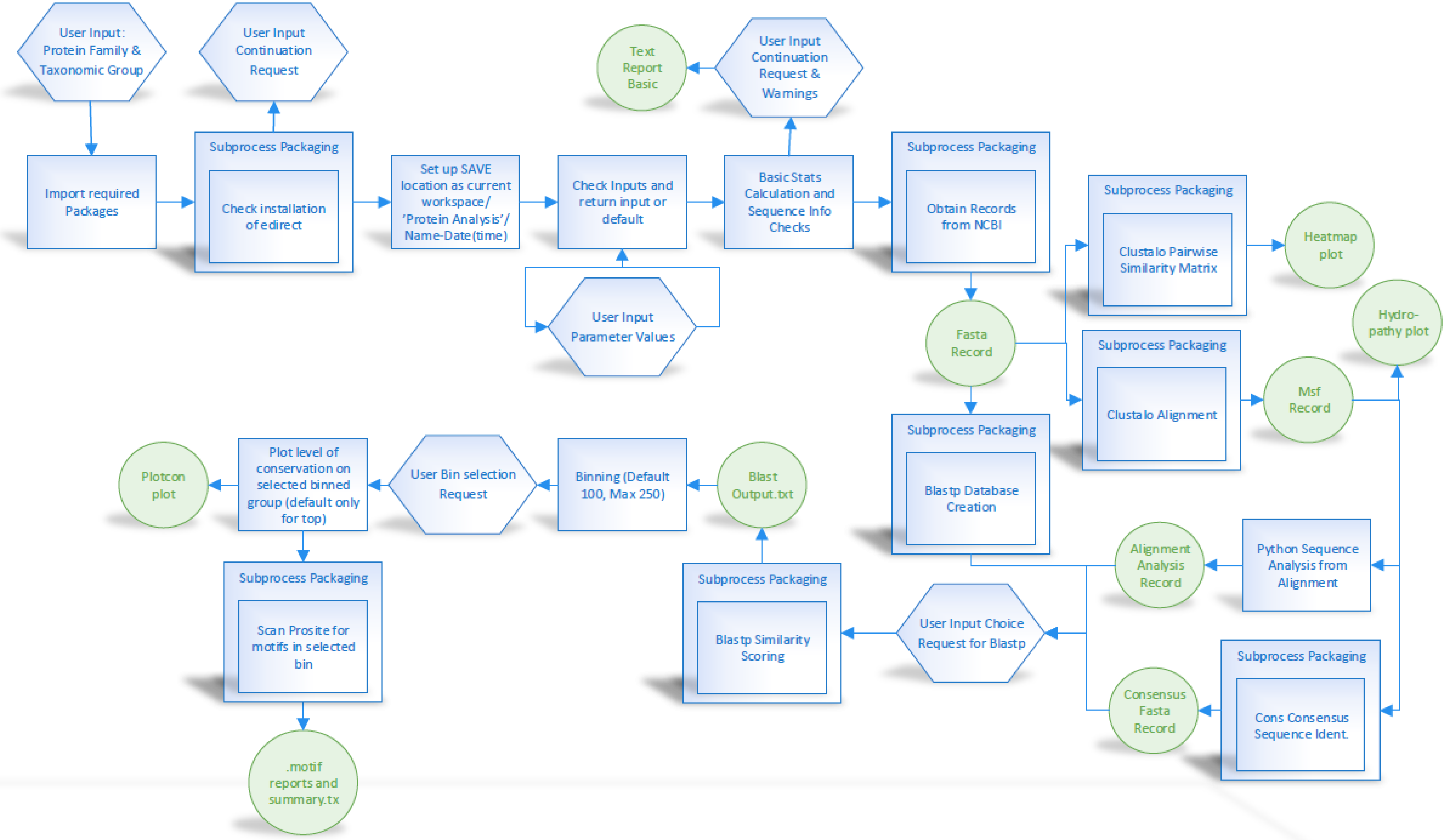
FOLDER NAME	WORKSPACE LEVEL	FOLDER SPACE	DESCRIPTION
Protein_Analysis_[name] _[date]_([time])	0	Home space or specified	Working directory space for analysis
Motifs	1	Protein_Analysis_[name] _[date]_([time])	Folder container for all bins
Bin_[bin_number]	2	Motifs	A separate folder for each blastp rankings bin selected for further processing. Stores individual fasta sequences and motif files for any sequence_id in the given bin_number.

Table of File Outputs :

FILE NAME EXAMPLES	DESCRIPTION
G6pc_txid8782_seqs.fasta	Fasta file containing all sequences pulled using taxid/family group name
NCBI_retrival_summary.txt	Summary of data pulled from NCBI with basic info
G6pc_txid8782_clustal.msf	Clustalo alignment file used to produce a consensus sequence from all given fasta files.
G6pc_txid8782_consensus.fa	Consensus sequence produced from clustalo alignment. Can be used to blast against custom fasta database.
G6pc_txid8782_blastdb.phr	Blastdb database file used for blastp processing of query sequence against custom database of given fasta sequences.
G6pc_txid8782_blastdb.pin	Blastdb database file used for blastp processing of query sequence against custom database of given fasta sequences.

G6pc_txid8782_blastdb.psq	Blastdb database file used for blastp processing of query sequence against custom database of given fasta sequences.
G6pc_txid8782_consensus_blastp_output.txt	Blast output file showing relatedness between analysed sequences named with the query_file between [name] and _blast_output.txt
G6pc_txid8782_stats.pepstats	Statistic file containing summary statistics on proteins for all analysed fastas
Motifs/bin_0/XP_010411101.1.fasta	A individual fasta file for a sequence_id selected for processing based on the bin position calculated from blast results
Motifs/bin_0/XP_010411101.1.motif	A motif search report for a sequence_id for a given bin position calculated from blast results
G6pc_txid8782_similarityplot.svg	Sequence conservation between all processed fasta files
G6pc_txid8782_hydropathy_alignment.svg	A plot of the hydrophobicity over the the whole length of all processed sequences
G6pc_txid8782_align.txt	A alignment file generated through pairwise comparisons from clustalo
G6pc_txid8782_matrix.txt	A similarity matrix generated through pairwise comparisons from clustalo
Similarity_matrix_heatmap.png	A heatmap plot of the similarity between sequences as found by pairwise alignment with clustalo

The following page contains a diagram of the main processes and outputs involved in the protein_analysis.py script.


User Input: Protein Family & Taxonomic Group
Import required Packages
Subprocess Packaging
Check installation of edirect
User Input Continuation Request
Set up SAVE location as current workspace/ ’Protein Analysis’/ Name-Date(time)
Check Inputs and return input or default
User Input Parameter Values
Basic Stats Calculation and Sequence Info Checks
User Input Continuation Request & Warnings
Text Report Basic
Subprocess Packaging
Obtain Records from NCBI
Fasta Record
Subprocess Packaging
Clustalo Pairwise Similarity Matrix
Heatmap plot
Subprocess Packaging
Clustalo Alignment
Msf Record
Hydro-pathy plot
Subprocess Packaging
Blastp Database Creation
Python Sequence Analysis from Alignment
Alignment Analysis Record
User Input Choice Request for Blastp
Subprocess Packaging
Cons Consensus Sequence Ident.
Consensus Fasta Record
Subprocess Packaging
Blastp Similarity Scoring
Blast Output.txt
Binning (Default 100, Max 250)
User Bin selection Request
Plot level of conservation on selected binned group (default only for top)
Plotcon plot
Subprocess Packaging
Scan Prosite for motifs in selected bin
.motif reports and summary.tx

MAIN TASKS NOT INCLUDED IN BASE MANUAL:

All functions have a docstring which should direct the user to the appropriate type, indicate default values and provide a description of the variable's content/use process.

The main parent function master_analysis() provides a basic use scenario, with all inputs for the rest of the pipeline generated either by functions, using the keys in the default dictionary or by user inputs. To use the sub-task master functions parameters and default values will have to be entered during the function call.

Some processes have not been described in the basic overview:

- Checking edirect installation
- User string parsing
- Error checks
- Function inputs

IMPROVEMENTS

The script is currently suboptimal due to time limitations. The following additional functionality should be added:

- Ability to enter text of gene families rather than specified lists of genes
- Ability to automatically search for gene alias's
- Ability to enter taxonomic ids without the prefix txid
- Ability to enter taxonomic group names and find the taxonomic ids
- Kwargs into non_python_processes to allow greater flexibility in commands & output formatting
- Addition of pepinfo plots (current makes files for parsing into charts but does not save chart output)
- Plotting options per bin using the blastdb and fastadb outputs
- Change storage method so that each sub-process master function has it's own specified folder.
- Change storage method so that interrupted runs have an option to delete partial results before sys.exit is called.
- Change default value storage and method of passing into functions to include a description for the ease of the user.
- Additional protein statistics analysis using both python processing and emboss Unix commands
- Addition of 'add_string' variable to add specific inputs to run_nonpython_process() function if they are not hardcoded into the current functions.

MOST IMPORTANT CHANGES:

- Test cases to ensure all variations of use work correctly
- User input parsing to convert string input back to correct typing (without relying on escaped quotes or using pairs of ' and " together.
- Parsing of kwargs!