**Comparative Genomics Practical 07**

**Function Prediction**

**Group 06 Xueqing Wang Tianlin He**

**Summary**

This exercise can be divided into three parts: functional annotation of each protein in a proteome, topology prediction of transmembrane protein and subcellular localization analysis. Three distinctive programs were used for these purposes: namely HMMscan which search for gene family in Pfam database; Phobius which tells us the fraction of transmembrane segments and signal peptides present in a query sequence; and finaly TargetP, which predicts the post-translational subcellular location of an eukaryotic protein based on the type of signal peptide it carries. These three softwares all powerful bioinformatics tools in protein functional study, as altogether they provide critical information of a unknown protein, even if only its sequence is available.

**Domain annotation**

Find the Pfam domain organization for the first 100 proteins encoded in your genomes.

1. The way to do this is to use the hmmscan program.

2. Easiest is to run it as

# Normally you would download library from Pfam

wget ftp://ftp.ebi.ac.uk/pub/databases/Pfam/current\_release/Pfam-A.hmm.gz

# Each Pfam file is described by release notes

# ftp://ftp.ebi.ac.uk/pub/databases/Pfam/current\_release/relnotes.txt

# Than you decompress is and prepare it for hmmscan tool

gzip -d Pfam-A.hmm.gz

hmmpress Pfam-A.hmm

# We have done that for you already, and the Pfam/ files are provided in

course directory

# Run the hmmscan

hmmscan --cut\_ga --acc <hmm database file> <query protein file>

# Where hmm database file is Pfam-A.hmm and query protein file is your

proteome in multi-fasta format

a. What do the options mean?

In the terminal, run

hmmscan -cut\_ga -acc -o out.03hmm.txt Pfam-A.hmm protein100.03.fa

--cut\_ga : use profile's GA gathering cutoffs to set all thresholding

--acc: prefer accessions over names in output

-o out.03.gmm.txt: direct the output of this comment to a file named “out.03hmm.txt”

b. As database file, use​ Pfam/Pfam-A.hmm. ​ This may take a while.

c. To parse the results, you can use the command

cat hmmscan-output.txt | perl hmmscan\_parser.pl

we use the command

hmmscan out.03hmm.txt | perl hmmscan\_parser.pl >function.03hmm.txt

so that the output from hmmscan\_parser.pl can be directed to a file named “function.03.hmm.txt”, instead of showing on screen.

The output file function.03.hmm.txt is in attachment no. 1.

3. What kind of output does this give?

A typical line in the output file “function.03.hmm.txt” can be

./03.fa.txt\_orf00008 PF00180.19 Isocitrate/isopropylmalate dehydrogenase 1.4e-114 1 346 4 344

which displays the name of the protein we search, the family it belongs to, and its protein ID and E-value etc. These results are generated by matching the sequence of query protein with known sequences/families in Pfam database, using a hidden Markov Model.

4. Using the ​ pfam2go​ map (http://geneontology.org/external2go/pfam2go), assign gene

ontology terms to each of the genes.

a. Best is if you write a script to do this. If you do not, you can use

b. pfam2goTransfer.py​ which takes as arguments the pfam2go file and the

output from​ ​ hmmscan\_parser​ .pl​ . If you do use this, you must answer the

following questions:

We first downloaded pfam2go using the commend:

wget <http://geneontology.org/external2go/pfam2go>

and then run the pfam2goTransfer.py by taking the two arguments: pfam2go file and the output “function.03hmm.txt” from the previous task.

python2 pfam2goTransfer.py pfam2go function.03hmm.txt

i.How many dictionaries are used in the program? What is gained by using

them in the places where they are used?

It contains three dictionaries, namely pfam2go, lDomains, and lPredictions.

- pfam2go extracts information from the pfam2go file we downloaded from the Internet, so that it generates a dictionary with protein ID as a key and its corresponding GO term as item. Such as:

'PF05118': 'GO:0018193'

- lDomains extracts information from our result generated by the hmmscan, as it reads the file “function.03hmm.txt”. It simply creates a dictionary containing IDs of all proteins appear in the file when it reads a line in the file, such as:

{'PF07661': ''}

- lPredictions searchs for overlapping in key between pfam2go and lDomains, the ultimate goals is to display the proteins in “function.03hmm.txt” together with its corresponding GO term and function on a single line.

ii.What are the purposes of the third and fourth split commands?

The third split for domain in arch.split (): removes the number after “.” in a protein ID. For example, PF00326.20 becomes PF00326, so that it can match with those from pham2go file.

The fourth split for aGO in pfam2go [domain].split ("\t"): extracts the value in pfam2go dictionary separated by tab, so that it only checks if the ID of a protein from “function.03hmm.txt” matches with the GO term, but not the whole value containing GO, gene family, E-value etc.

iii.What type of variable is arch? What is its biological meaning?

Arch is the proteinID and the family it belongs to, such as :

PF00294.23 pfkB family carbohydrate kinase

It refers to architecture of a protein.

5. Do the results from what you ran above differ from simple BLAST (hypothetically

domain sequence vs proteome file)? How, why and to what extent.

In this practical, we ran the protein search with the use of profile Hidden Markov Model, which should be more sensitive than BLAST.[1] Besides, Pfam is not only a proteom database, it encompasses also information about protein family, the whole hierarchy and so on. Therefore, it gives more detailed information about the protein.

**Simple one-gene analysis using Phobius**

6. We will use Phobius, a fast and accurate predictor of TM topology.

Before you start the genome analysis, make sure that Phobius works from the command line. Make a test file called Q8TCT8 with the sequence of Q8TCT8 (see software documentation http://phobius.sbc.su.se/instructions.html).

/afs/pdc.kth.se/home/e/erison/Public/bin/phobius/1.01/phobius Q8TCT8.fa

Then run it on the web server. The results should be the same.

We run Phobius both in the command line and on the website, and got exactly the same predictions about the topology of Q8TCT8 sequence. The results are shown in AnnexI, which includes both the long and the short output format.

**Whole-proteome analysis with Phobius**

7. Now we want to run Phobius for all proteins in a genome. This assumes you have a fasta file with all proteins in each proteome from previous practicals. We need a script that launches Phobius for each sequence, and parses the output of Phobius. This is a very typical script in bioinformatics so it is a very general exercise. All we want to collect for now is the number of predicted signal peptides and TM (predicted transmembrane segments) segments for each protein, and find out for one proteome:

a. The fraction of proteins with 0 TM segments.

b. The fraction of proteins with > 0 TM segments.

c. The average number of TM segments for those with >0 segments.

d. The fraction of proteins with > 0 signal peptide.

e. The fraction of those (with > 0 signal peptide) with > 0 TM segment.

Tips:

You can call Phobius for each of sequence, or simply run

phobius -short <proteome file>

to get a one-line summary for each protein that is easily parsed without BioPython.

We first run phobius for each proteome in command line, and exported the short output into a txt file (one output file for each fasta file). Then we wrote a script to parse these output files and calculated the statistics outcomes required for this practical. The script is described in AnnexII. The outcomes are as follows:

Species 03 Proteome

The fraction of proteins with 0 TM segments in species 03= 0.7887796691440901

The fraction of proteins with > 0 TM segments in species 03= 0.21122033085590985

The fraction of proteins with > 0 signal peptide= 0.3176696235914649

The fraction of those (with > 0 signal peptide) with > 0 TM segment= 0.0890566037735849

The average number of TM segments for those with >0 segments= 4.545970488081725

Species 08 Proteome

The fraction of proteins with 0 TM segments in species 08= 0.784219001610306

The fraction of proteins with > 0 TM segments in species 08= 0.21578099838969403

The fraction of proteins with > 0 signal peptide= 0.11701556629092862

The fraction of those (with > 0 signal peptide) with > 0 TM segment= 0.15137614678899083

The average number of TM segments for those with >0 segments= 4.355721393034826

Species 09 Proteome

The fraction of proteins with 0 TM segments in species 09= 0.7972947585947774

The fraction of proteins with > 0 TM segments in species 09= 0.2027052414052226

The fraction of proteins with > 0 signal peptide= 0.14221303776066127

The fraction of those (with > 0 signal peptide) with > 0 TM segment= 0.21400264200792601

The average number of TM segments for those with >0 segments= 5.537534754402224

Species 18 Proteome

The fraction of proteins with 0 TM segments in species 18= 0.7805328409390662

The fraction of proteins with > 0 TM segments in species 18= 0.21946715906093378

The fraction of proteins with > 0 signal peptide= 0.11026114481667107

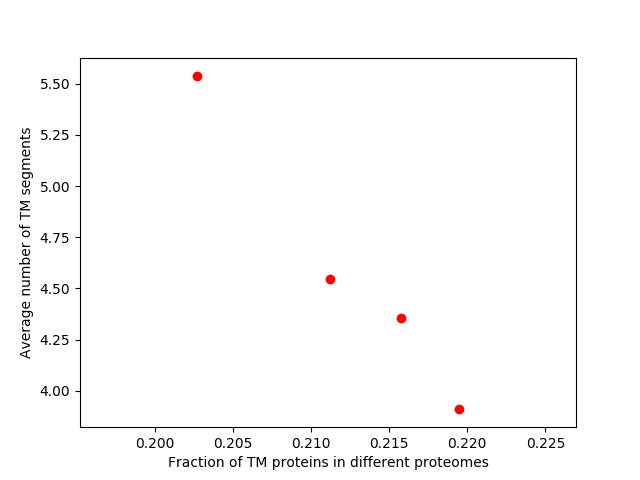
The fraction of those (with > 0 signal peptide) with > 0 TM segment= 0.22009569377990432

The average number of TM segments for those with >0 segments= 3.9122596153846154

**Comparative proteome analysis with Phobius**

8. Now run the previous analysis on all your (real) genomes. Make an xy scatter plot showing the fraction of TM proteins on one axis and the average nr of TM segments on the other axis. Is there a trend?

We collected the fractions and average numbers of transmembrane segments obtained in the last section, and made a scatter plot with average nr against fraction of TM proteins. The plot is shown as follows:

From the plot we could tell that, when the fraction of proteins with > 0 TM segment increases in the proteome, its average number of transmembrane segments in the transmembrane proteins drops. Their relationship is roughly linear.

**More protein localization analysis with targetP**

TargetP can predict more subcellular localizations, namely mitochondrial (only eukaryotes) and chloroplast (only for plants). Test it on your yeast chromosome via: http://www.cbs.dtu.dk/services/TargetP/

9. What does Plant/Non-Plant parameter do?

TargetP predicts the peptide locations based on chloroplast transit peptide (**cTP**), mitochondrial targeting peptide (**mTP**) and secretory pathway signal peptide (**SP**). If we select Plant parameter, it would detect throughout the submitted sequences with all these three signals; if we select non-plant it would only predict against mTP and SP. Since we are testing on yeast genomes we only need the Non-plant prediction.[2]

a. What fraction are predicted mitochondrial proteins?

In total there are 248 protein sequences in the proteome of yeast predicted by Genscan. When we run TargetP on the website without setting a cutoff, 29 peptides are predicted as mitochondrial proteins, taking of 11.7% of the total number. However, among the predicted mitochondrial proteins only 10 have relatively high reliability (difference between being interpreted as mitochondiral and the type with the second highest probability larger than 60%). To improve the quality of the predicted topology, we set the cutoff into 0.95, in which way only 12 proteins were considered as mitochondrial in TargetP.

b. How many are both predicted mitochondrial and to have a signal peptide?

(Comment on such predictions, are they biologically sound?)

All the predicted mitochondrial protein sections have a probability to be signal peptide; however, most of these probabilities are extremely low compared to the probability of being in mitochondron.

Both the mitochondrial and signal peptide proteins have biological meanings. Mitochondrial proteins are targeted into mitochondron by its targeted part in the N-terminal, which is cleaved after it fulfilled its mission (leading the peptide into mitochondrial). The peptide will then stay in the mitochondrial and perform function in it. On the other hand, signal peptide is utilized by the secretory protein to enter the secretory pathway[3]. Secretory proteins can be found in the ER membrane (smooth or rough), ER lumen, as well as within the cytoplasm matrix and extracellular sites as they are secreted into different places. Thus, mitochondrial proteins and secretory signal peptides are totally distinctive both in localization and function, and the prediction of both mitochondrial and signal peptide makes no sense in practical.

10. Would your run targetP for all of your genomes? Why?

We would not run TargetP for the prokaryote genomes, as the location assignment is based on the predicted presence of any of the N-terminal presequences: chloroplast transit peptide (**cTP**), mitochondrial targeting peptide (**mTP**) or secretory pathway signal peptide (**SP**). All these three peptide fractions are only found in eukaryotes as they have the corresponding organallas. If we test prokaryote genomes based on these detections it would make no biological sense.

**Discussion**

Although HMMscan, Phobius and TargetP work on different aspects in protein analysis, they have some shared features which make them useful bioinformatics tools. First of all, they only require protein sequence as input, such as a single-fasta format file of a protein, or even a multi-fasta file of a proteome. In this practical, the multi-fasta proteomes generated from GLIMMER or GENSCAN were taken as input to these programs.

As mentioned, HMMscan is both reliable and sensitive in protein family annotation as it is based on the probabilistic hidden Markov model as well as the enormous database in Pfam-A. With the use of a few python scripts, we can easily parse the result from HMMscan, which gives us a comprehensive view of function, protein family, E-value of genes detected by GLIMMER/GENSCAN. It is also noticed that some genes still remain un-annotated, which is possibly due to 1) the prediction from GLIMMER/GENSCAN is not accurate enough that it poorly resembles any known protein sequences in Pfam database; 2) the similar protein exist in Pfam, but its protein family/function has not yet been elucidated. [1]

In addition to analyzing the biophysical properties of amino acids in the protein sequence, Phobius is based on the hidden Markov model built on transmembrane protein topology. Hence, it dramatically increases the accuracy and decreases the error rate of topology prediction than other traditional methods, such as TopPred. [4] In this practical, statistics of fraction of transmembrane segments in each proteome were collected. By plotting a x-y plot using these data, it can be concluded that the number of transmembrane segment drops linearly as there is higher percentage of transmembrane proteins in the proteome.

Compared with Phobius, TargetP is even more specialized in predicting signal peptide in proteins. It tells us which cellular compartment would a protein probably end up with, according to the short, highly hydrophobic signal peptide sequence it carries, which is a unique feature for eukaryotes. [3] In this practical, we further learn that plant and non-plant species have their own way in defining a signal peptide, which should be clearly differentiated when we conduct the search.

**Reference:**

1. RD Finn, J Clements, W Arndt, BL Miller, TJ Wheeler, F Schreiber, A Bateman, SR Eddy. 2015 HMMER web server: 2015 update. *Nucleic Acids Research* 43: 30-38.
2. O Emanuelsson, H Nielsen, S Brunak and G von Heijne. 2000.Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol.* 300: 1005-1016,
3. H Nielsen, J Engelbrecht, S Brunak and G von Heijne. 1997*.* Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering* 10:1-6.
4. L Käll, A Krogh, EL Sonnhammer. 2004. A combined transmembrane topology and signal peptide prediction method. *Journal of Molecular Biology* 338: 1027-1036.