**Phylogenetic Aware Parsing Script**

‘Phylogenetic Aware Parsing Script’ (PAPS for short) is the last of a pipeline of three Perl scripts which allows the user to produce lists of homologous genes (proteins clustered using MCL) based on their taxonomic occupancy; this allows the user to infer the patterns of gene gains and losses across a phylogeny which can be customized by the user. This file describes the scripts and series of steps to follow to run the pipeline. Most instructions assume that the pipeline is being used in a Unix environment and the user has some basic programming knowledge. The scripts are thoroughly commented, and provide usage instructions when run. Please, check that the scripts here provided have executing permissions in your system.

After obtaining a list of homologous genes (HG) using BLAST1 together with MCL2, the pipeline processes the MCL output (a table in which each row is a HG, and each cell contains the FASTA header of each sequence belonging to that HG) and produces the files needed later for PAPS. PAPS has a user friendly command-line interface which can be used to obtain lists of HG based on a set of criteria defined by the user, such as presence or absence of the HG in different clades at any phylogenetic level defined by the user. This allows the user to tailor very flexible queries that are processed by PAPS as required; for example, in a dataset containing genomes from the Animal Kingdom and other eukaryotes, the program can produce lists of HG present in all Metazoa and absent in the rest of the taxa (metazoan novelties), or HG present in all Arthropoda minus one species, or HG present in *Homo sapiens* and at least 2 Hexapoda but absent in all Lophotrochozoa, etc.

A description of the dataset used during the assembly of this pipeline can be found in the paper by Paps & Holland3. The assembly of this pipeline was finished in the Department of Zoology (University of Oxford) in 2015 with help of Patrick Gemmell (Oxford) for the *PAPS.pl* subroutines involving a hash of hashes. This manual was written in the School of Biological Sciences (University of Essex) in June 2017.

1. **Preparation**: download the protein complements for the genomes of interest. Modify the FASTA headers to include a short label indicating the species to which the gene belongs. The label should be at the very beginning of the FASTA header. For example:

>Dmel\_CG0000008\_a\_arc\_Source\_FlyBase\_0000008  
>Dmel\_CG0000014\_abd\_A\_abdominal\_A\_Source\_FlyBase\_0000014  
>Dmel\_CG0000015\_Abd\_B\_Abdominal\_B\_Source\_FlyBase\_0000015  
>Dmel\_CG0000017\_Abl\_Abl\_tyrosine\_kinase\_Source\_FlyBase\_0000017  
  
A list of these short labels should be produced by the user. We suggest to introduce the labels sorted by phylogenetic relationships (e.g. all vertebrates together, all reptiles together, etc.) as this makes later steps and the interpretation of the results much easier. We would also like to advise to check rest of these instructions, and the example tree and the hash of hashes described in step 5 before producing this list of short labels.

The list of labels should be introduced in the array in line 81 of *MCL\_row\_counter.pl* (within folder “*Input*”), and similarly in lines 35 and 229 of *PAPS.pl*.

We recommend to include descriptions of the genes or gene symbols in the FASTA header in as many genomes as possible, as this makes much simpler to identify the genes found in a given HG. Genomes with FASTA headers annotated like this should be indicated in *PAPS.pl*, lines 471 and 499.

1. **BLAST and MCL**: run BLAST and MCL, following the instructions found in the later (BLAST section in [MCL web page](https://micans.org/mcl/index.html)).

Briefly, the user should create a BLAST database containing all the proteomes to be compared using [makeblastdb](https://www.ncbi.nlm.nih.gov/books/NBK279688/). Then, in order to perform an all vs all comparison, a BLAST search should be performed using the very same proteomes as query against the database. The proteomes can be merged in a single query file, or each proteome can be BLASTed independently against the database; the latter is recommended to speed up the process in a multiprocessor system, as each BLAST search can be submitted to a different processor, which is much faster than using the multiple threads option implemented in BLAST.

The user must decide an e-value threshold for the BLAST search (e.g. -evalue 0.00001), and specify the tabular output (-outfmt 6). An example of such BLAST search:

blastp -query *proteome1* -db *all\_proteomes\_db* -evalue 0.00001 -outfmt 6 -out *blast\_output*

If there are multiple output files containing the results of the BLAST search, then they should be merged in a single file (for example, suing the Unix command cat). MCL needs to prepare the output file of BLAST with mcxdeblast before performing the actual gene clustering with the program mcl. Before running the program mcl, the user must decide which inflation value (-I) to use, an ideally test different inflation values. In doubt, use the default (2.0) which was the best value for our dataset. This is an example of the commands used with the MCL programs, more details can be found in the [MCL webpage](https://micans.org/mcl/index.html):

mcxdeblast --m9 --line-mode=abc --out=*mcxdeblast\_output* *blast\_output*

mcl <*mcxdeblast\_output*> --abc -I 2 -o *mcl\_output*

1. ***MCL\_row\_counter.pl***: this script will parse the output of MCL to produce a taxonomic occupancy table. Place the MCL output file, as well as a FASTA file containing all the protein sequences of all the genomes, in the “*Input*” folder. Execute *MCL\_row\_counter.pl*, which will parse the output of MCL and produce a new tabular file. In this file, each row is a HG, each column a species, and the numbers in the cells indicate how many sequences of that species are present in that HG.
2. ***Create\_DBs.pl***: leave the output file of *MCL\_row\_counter.pl* in the “*Input*” folder and return the main folder containing *Create\_DBs.pl*. This script will create Perl databases to speed up the *PAPS.pl* searches. Before running *Create\_DBs.pl*, modify lines 9, 10 and 11 to match your file names. After running this script, check the permissions in the resulting database files and make them available to all users.
3. **Preparing for *PAPS***: before running *PAPS.pl*, some modifications are needed:
   * 1. Double check the edits to *PAPS.pl* specified in step one.
     2. Modify file names in lines 12 and 13 to match your file names.
     3. The most complicated part: implement the phylogeny of your groups of interest in the subroutine HASH\_SPP (line 590). This subroutine contains a hash of hashes, each line contains one species and its classification levels (e.g. {'Opisthokonta'}, also see lines 680 to 694), followed by an index. For example:

$spp{'Eukaryota'}{'Amorphea'}{'Opisthokonta'}{'Holozoa'}{'Node1'}{'Node2'}{'Metazoa'}{'Eumetazoa'}{'Planulozoa'}{'Bilateria'}{'Deuterostomia'}{'Chordata'}{'Olfactores'}{'Vertebrata'}{'Homo\_sapiens\_(Hsap)'}{''} = 61;

Lines can be added and removed (e.g. add or remove a vertebrate), but all lines must keep the same number of classification levels (designated by pairs of curly braces); empty classification levels ({''}) can be used.

**IMPORTANT!!!** **The index number MUST CORRELATE with the position of the species in the array of species declared in step 1**; therefore, the first species in the array of species (step 1) should have the index 0, the next one index 1, etc. The file *Tree.txt* shows the example tree that was used in our dataset; we advise the users to generate a similar tree with indexes (pen and paper will do!) as a guideline to produce the list of labels detailed in step 1.

1. **Running *PAPS.pl***: execute *PAPS.pl*. Depending on the size of your dataset it may take a few seconds to read it. Some information of your dataset will be printed on the screen, followed by the phylogeny introduced in step 5. This tree can be invoked any time by typing “tree”. Finally, a command prompt will ask user which criteria should be used to mine lists of HG. Examples of queries can be seen by typing “example”:

Clade/species names can be truncated, but the start of the clade name should match the table printed above.

Search is case insensitive.

Some search examples (first 4 digits in examples stand for rest of taxa, the other 4 for ingroup):

"Vertebrata-present" => genes found in ALL vertebrate species, present or absent in other clades/rest of taxa

Rest of taxa ???? Ingroup 1111

"Vertebrata-present Rest-absent" => genes found in ALL vertebrate species, absent in other clades/rest of taxa

Rest of taxa 0000 Ingroup 1111

"Vertebrata-present Rest-present" => genes found in ALL vertebrate species, present in other clades/rest of taxa

Rest of taxa 1111 Ingroup 1111

"Vertebrata-absent Rest-present" => genes found in rest of taxa species, absent in Vertebrata

Rest of taxa 1111 Ingroup 0000

"Homo-present Mus-present Rest-absent" => genes only found in humans and mice. Species can be specified one by one.

The number of species presenting/missing for a gene can be fine-tuned with minus#, atleast#, only# for both ingroup and rest of taxa:

"Vertebrata-minus1" => found in ALL vertebrate species but one, present or absent in other clades/rest of taxa

Rest of taxa ???? Ingroup 1110 / 1101 / 1011 / 0111

"Vertebrata-minus2 Rest-minus1" => genes found in ALL vertebrate species but one, absent in other clades/rest of taxa

Rest of taxa 1110 / 1101 / 1011 / 0111 Ingroup 1100 / 1010 / 1001 / 0110 / 0101 / 0011

"Vertebrata-atleast1 Rest-atleast1" => genes found in at least 1 vertebrate species and 1 rest of taxa species

Rest of taxa 1000 / 1100 / 1110 / 1111 / 1010 / 1011 / 1001 / 1101 / 0110 / 0111 etc.

Ingroup 1000 / 1100 / 1110 / 1111 / 1010 / 1011 / 1001 / 1101 / 0110 / 0111 etc.

"Vertebrata-only3" => return genes found in just 3 vertebrate species, present or absent in other clades/rest of taxa

Rest of taxa ???? Ingroup 1110 / 1101 / 1011 / 0111

Different criteria can be combined in a single search:

"Vertebrata-minus1 Echinodermata-atleast2" => genes found in ALL vertebrate species but one, AND present in at least two echinoderms, absent/present in other clades/rest of taxa

Rest of taxa ???? Vertebrata 1110 / 1101 / 1011 / 0111 Echinodermata 1100 / 1010 / 1001 / 0110 / 0101 / 0011

"Vertebrata-atleast2 Urochordata-atleast2" => genes found in 2 or more vertebrate species OR 2 or more urochordates, independently if they are present/absent in other clades/rest of taxa

Rest of taxa ???? Vertebrata 1100 / 1010 / 1001 / 0110 / 0101 / 0011 Urochordata 1100 / 1010 / 1001 / 0110 / 0101 / 0011

"Nematoda-absent Platyhelminthes-absent Rest-present" => genes found in clades/rest of taxa, absent (convergently lost) in round worms and flatworms

Rest of taxa 1111 Nematoda 0000 Platyhelminthes 0000

Careful with nested taxa!!! Start with the greater group taking into account the conditions for the smaller group:

To find genes in ALL chordates but missing only in humans => "Chordata-minus1 Hsap-absent"

To find genes in ALL chordates but missing only in vertebrates => "Chordata-minus5 Vertebrata-absent"

To find genes in at least one clade of chordates, but missing only in vertebrates => "Cephalocordata-atleast1 Urochordata-atleast1 Vertebrata-absent"

After performing the search, information the findings will be printed on the screen. Then the script will ask the user if the results should be saved. In that case, four output files will be produces in the folder “Output”, indicating the query used and number of HG found in the file name. For example:

Metazoa-atleast43\_Outgroup-none\_25\_HGs\_MCL\_annotated\_genes.out

Metazoa-atleast43\_Outgroup-none\_25\_HGs\_MCL\_columns\_parsed.out

Metazoa-atleast43\_Outgroup-none\_25\_HGs\_MCL\_genes\_IDs.out

Metazoa-atleast43\_Outgroup-none\_25\_HGs\_taxa\_names.out

The file “…\_MCL\_annotated\_genes.out” contains, for each HG, the FASTA headers of the genomes that are indicated as annotated (see step 1, headers with gene symbols or descriptions). The file “…\_MCL\_columns\_parsed.out” contains the occupancy table for all the HG, this file can be opened with Excel (tabulators are the field delimiters). The file “…\_MCL\_genes\_IDs.out” cointains, for each HG, all the FASTA headers of all the genes and genomes included in each HG (not only the headers coming from annotated genomes). Finally, the file “…\_HGs\_taxa\_names.out” contains the list of taxa present in each HG.

You can terminate PAPS when you are done performing searches.

And that’s it! Feel free to contact me ([jpapsm@essex.ac.uk](mailto:jpapsm@essex.ac.uk)) if I can be of further help.

**References**

1. Camacho, C. *et al.* BLAST+: architecture and applications. *BMC Bioinformatics* **10,** 421 (2009).

2. Enright, A. J., Van Dongen, S. & Ouzounis, C. A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30,** 1575–1584 (2002).

3. Paps, J. & Holland, P. W. H. What makes an animal? *J. Placeholdering* (2017).