--- PhyloMatch package ---
User Manual
matteo.ramazzotti@unifi.it

Data preparation

- Download genomic data from Sanger, NCBI or SGD. Both .gff and .fa files are needed. The .gff file is a file format used for describing the position of genes on chromosomes. The .fa file is a fasta file containing the sequence of each chromosome.
- Use the appropriate script for creating the main genomic files with CDS sequences (e.g. cds_catch_xxx.pl, where xxx = ncbi, sanger or sgd). Due to differences in the organization of the three centers, three different scripts are needed. We also have an auto-launcher that process all the genomes in a folder. In the end a number of files are created with extensions like
  - .cds.txt (the coding sequences)
  - .prot.txt (the protein sequence obtained by translating the corresponding cds in the cds file and
  - .log with a report of the extraction, included errors, features of the cds and etc.

The most important features of these files are

- the names of files are taken from the organism name
- in each file the fasta headers contain gene names that are shared across the different files. What we want here is to easily map the same gene in all the available organisms.

Alternatively, some genome is available in the processed form from the PhyloMatch website.
Put all the .cds.txt files in the same folder named “genomes” from which the main script will read them by default.

Package work-flow

Check the presence of a given gene in all the organisms and create a list of universally present genes. Then, for each gene:

1. Create a fasta file containing all the sequences of a given gene in all the organisms
2. Align the sequences in the file.
3. Optionally apply clipping of extremities
4. Optionally create SNPs of the aligned sequenes.
5. Optionally concatenate the alignments
6. Compute a phylogenetic tree
7. Optionally bootstrap the tree
8. Create a cluster file containing the descendant of all the nodes of the tree.
9. Search for a given target on the tree (e.g. a triplet and a couple of organisms that cluster together)

The point n.8 is ideally distinct from the previous steps and can be delayed until all the trees and clusters have been created. This is how the package has been intended for. In fact, the first 7 steps can be parallelized to speed up cluster collections. In addition, curious people can launch the search while the first 7 steps are running.

The scripts

phylo_match.pl: this is the main script. It check for data integrity and perform the analysis from sequence collection to tree generation, including the concatenation of alignments if gene couples, triplets (or k-tuples in general) are required. It is a glue between several programs (e.g. clusterw, phylip) and scripts (see below). It also copes with the parallelization.

snipper.pl: this script analyzes an alignment and emit only the columns that contain some variations. Despite of the name of the script, every variation is taken into account, so if a large deletion is present in one of the genes, all that region will be present in the output.

clipper.pl: this script analyzes the extremities of an alignment and trim it to the sequence that starts after and to the sequence that stops earlier. The alignment body is reduced and more consistent.

tree_rename.pl: since the intermediate alignment and phylogenetic formats does not allow long sequence names, and since organism names are used to create trees, sequences are initially encoded with numbers, the correspondences written to a file named strains.txt and in the end this script relabels the tree with original names.

phylo_screen.pl: this is the script that scans the trees and the clusters and identify the gene or combination of genes that match the target required by the user.

Configuration of the analysis

The configuration of the analysis (the first 7 steps) is written (and can be accessed and modified) in the file named phylo_match.conf. All the options must be specified in the form key=value. Remmed (#) or blank lines are ignored. Here the details of the different options are defined:

- **mode = [mono|couple|triplet|genome|random|list|query]:** mono means that single genes are used, couple means that concatenations of couples of genes are used, triplets means concatenations of triplets of genes are used, genome means that all the available genes are concatenated, random means that a specified number of random k-tuple concatenations of genes (see below) are used, list means that a list of possibilities have to be specified in a file, query means that all the genes specified in a file (or at supplied at the command line) will be concatenated and used.

- **rep = NNN**: used if random mode is on, specify how many random combinations have to be tested
- **rep_n = N**: used if random mode is on, specify how many genes have to be concatenated.

- **use_snip = [1|0]**: apply or not the information reduction on alignment column (snipping)
- **use_clip = [1|0]**: apply or not the trimming of the alignment body on the most valuable region, i.e. that shared by all the organisms (clipping).
- **use_boot = [1|0]**: only used if use_snip=0, i.e. if an analysis is based on both variable and constant positions, specify whether or not to apply bootstrap to a tree
- **tree_mode = NJ**: only neighbor-joining is supported by now.
- **acc_file = filename**: a file where to write already processed elements. This is useful if, for some reason, an analysis is interrupted and have to be resumed.

- **selfile = filename**: a list of genes to be analized. It is a way of restricting the whoel analysis to a list of interesting genes rather than to all the genes of a genome.
- **checktree = [1|0]**: skip the analysis of an element for which a tree is already been computed (present in the appropriate folder)
- **check_acc = [1|0]**: whether to use or not the selfile specified above.
- **no_phylo = [1|0]**: this is a trick. If on, no phylogenetic analysis will be preformed but all the files will be prepared for it (alignment, concatenation, snipping, clipping etc.). It is like to say: just prepare the files, trees will be computed later.
- **already_prepared = [1|0]**: the second part of the trick. If all the files are ready, with this option on no alignment or other preprocessing will be skipped and only the phylogenetic analysis will be performed. Just be sure that all is really "already prepared".
- **processors = NN**: the number of processors to be used, so it determines the degree of parallelization. Please consider that parallelization here is thread-based via forking, no computer cluster support is present at this moment.
- **skip_missing = [1|0]**: if a gene is not available in a group of concatenated genes, strip that gene and proceed (1) or totally skip that analysis (0).
- **save_concats = [1|0]**: whether to save or not the concatenated alignments. Usually this is not required...
- **write_every = NNN**: if millions of trees have to be generated. The computer inodes will saturate, so every NNN analyses all the trees and clusters will be merged in a single file and the single files removed.
- **tree_loc = XXX**: the name of the folder where the trees will be saved.
- **cluster_loc = XXX**: the name of the folder where the clusters will be saved (usually similar to that of trees above).

One note about the combinatorial approaches proposed are needed. If you set up the script fo working in couple mode on a full genome let's say of 6000 genes, you have to expect ~ 18 million trees and clusters to be generated. Though parallelizable, this procedure could spend weeks to complete. In triplets mode the combinations are ~ 36 billions. Definitely too many... So please use a selection file when using couples or triplets mode to reduce the number of genes to be tested to a reasonable number.

Intermediate files and folders

The programs start by preparing a folder environment for the storage (and re-usage) of intermediate files. Here the list of folders is reported, basically sorted by their order of usage during the work-flow:
genomes: where the genomic files are placed
fastas: store the fasta formatted files containing the sequences of each gene shared by all the organisms. Files are named with gene names (gene_name.fasta) and sequence labeled with numerical indexes (that are traced back in the strains.txt file) to grant compatibility with following programs.
Alns: store the phylip formatted alignment of the fasta files contained in the previous folder (gene_name.phy).
Clips: store clipped versions of the alignments (gene_name.clip.phy)
snips: store snipped versions of the alignments (gene_name.snip.phy) or of their clipped versions (gene_name.clip.snip.phy)
trees: store files containing the nexus formatted trees resulting from the phylogenetic analysis of the alignments (gene_name.renamed.tre). The file names are marked with "renamed" because they does not anymore contain numerical indexes but true organism names. If required, several trees can be grouped in a single file to reduce the number of files created, that can be very large.

clusters: store files containing an enlarged textual representation of a tree. Basically all the nodes are written, one per line, in terms of all their descendants (leaves) and their bootstrap support if required. The file names are like gene_name.clu. AS for trees, several clusters can be grouped in a single file.

The trees and clusters folders may contain two possible sub-folders: snip and full. This is related to the type of analysis required, so e.g. the results of a snip-based analysis will be put in the snip folder. Feel free to rename these folders after a full phylo_match analysis have finished, new folders (again, named snip or full) will be created on next run.

During the work, phylo_match makes extensive use of temporary folder where the sequences are processed. This is especially important if a parallelization is required. Each analyzed entity (an alignment or a concatenated alignment) is enclosed in a dedicated folder with auto-generated unique names, processed, emitted in the appropriate output folder and the temporary folder automatically deleted.

The screening process.

The phylo_screen.pl script reads a target file and a list of cluster files and searches for trees presenting the specified target conformation. The trees can be millions, e.g. all the couples that can be formed by combining couples of 5000 genes. The target is specified like this:

name1
name2

name3
name4
name5

An example of a correctly formatted target file is available at Duccio Cavalieri's homepage. In this format an empty newline represents the end of a group, so the example above target one group of three (ideally) organisms AND one group of two organisms. They both have to be present exactly in a tree. This means that if a cluster file (representing a tree) presents a node ending at name1, name2 and name3 leaves and another node with name4 and name5 leaves, it will be emitted as “matched”. A tree with nodes containing name1 and name2 (not name3) leaves and another node with name4 and name5 leaves, it will not be matched. Please consider the intended stringency of this method: if the leaves of a node in the tree are name1, name2, name3 and name6 it does not exactly match the group 1, so this tree will not be matched.

The screener is very fast and allows to easily adapt the target file to match desired conformations. Since search is made on exact names, any spelling or mistyping or different case will cause the loss of a possible target. You are warned. A possible solution is to make a copy of the strains.txt file (that contain exact names) and simply modify the position of the element or remove them.