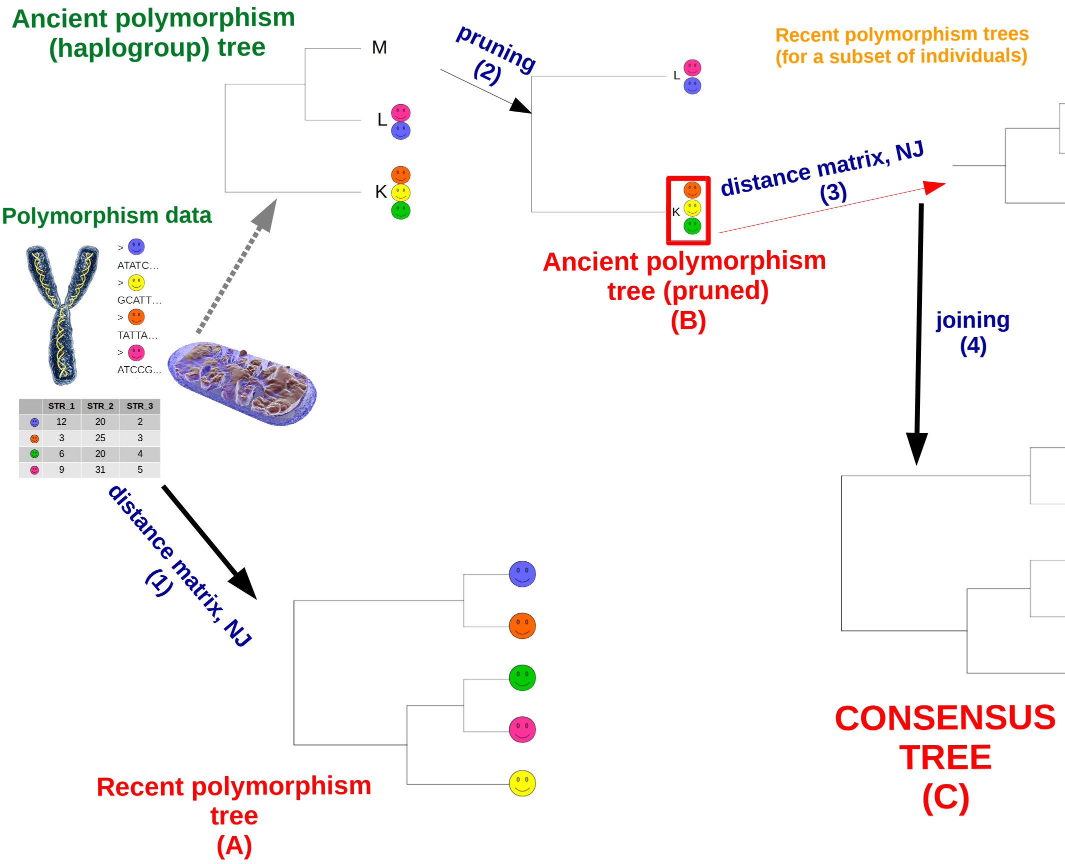
Materials and Methods

Available as both R and Python scripts, our program seeks to build a consensus tree that results from the merging of ancient and recent polymorphism phylogenies. Before digging into the implementation details, we begin this section by introducing the general structure of the program. Then, we supply some information about input and output data formats and finish with providing some explanations on tree construction.

General workflow of the program


Ancient polymorphism (haplogroup) tree
M
L
K
pruning (2)
L
K
Recent polymorphism trees (for a subset of individuals)
distance matrix, NJ (3)
Ancient polymorphism tree (pruned) (B)
joining (4)
Polymorphism data
>
ATATC...
>
GCATT...
>
TATTA...
>
ATCCG...
distance matrix, NJ (1)
Recent polymorphism tree (A)
CONSENSUS TREE (C)

	STR_1	STR_2	STR_3
🔵	12	20	2
🟠	3	25	3
🟢	6	20	4
🟣	9	31	5

Figure 1: *General workflow of the program.* Input data for this program (shown in green) are the recent polymorphism (STR or sequences) data as well as an haplogroup tree (pre-existing; dashed line). Tree construction involves two major steps: distance matrix calculation and aggregation of the taxa following the neighbor-joining (NJ) algorithm. First, a recent phylogeny based on the input data is built (1). The whole haplogroup tree is next pruned in order to obtain an ancient polymorphism tree that only contains the haplogroups present in the data set (2). Then, recent polymorphism trees (orange color font) are constructed for each subset of individuals that share the same haplogroup (3). These trees are joined at the corresponding tip of the haplogroup tree (4). The final result is hence a consensus tree of recent and ancient polymorphism phylogenies. In order of creation, the three outputed trees (indicated in red) are: the recent polymorphism phylogeny (A), the pruned haplogroup tree (B) and the consensus tree (C).

The workflow (Figure 1) is launched from the command line by typing for example (see readme file for complete description of the flag options):

```
user@host:~$ ./CARP.py -f seq -P seqData.fasta -H haploG.csv -T m
```

```
user@host:~$ ./CARP.R -v -f str -P allData.csv -T paternalHGtree.newick
```

Input and output data types

Since a large number of input formats were available, it was not trivial to design a program which is both user-friendly (i.e. quick launch) and flexible enough. After several trials, it turned out that the best option was to impose some restrictions on the input format. A readme file provides indications about the format that input data must have. Default parameters can also easily be changed in a separate script (e.g. presence or absence of a header, type of separator, etc.).

The program can work with recent polymorphism contained in STR or sequence file. STR data must be in a data frame - plain text (".txt") or spreadsheet (".csv", ".xls") - with a header and columns representing the loci, lines the individuals (individual identifiers must be in the first column and are treated as row names). Sequence data must be in FASTA format and are read using already implemented functions from the AlignIO Python module from Biopython (Chapman and Chang 2000) or from the seqinr R package (Charif and Lobry 2007).

Haplogroup information can be given either in the file containing the recent polymorphism data (for STR data only; haplogroups in the first column after row names) or in a separate file (individuals in lines and haplogroups in the first column). This latter must be inputed using the appropriate command line argument (see readme file). By default, however, the program provides the whole haplogroup tree extracted from the available phylogeny mentioned in the introduction.

Similarly, a large variety of output formats were also conceivable. We decided to restrict it to a single type of tree files, namely a Newick textual representation of the tree (".newick"). In order of creation, following phylogenies are outputed (with individual ID at the tips): the tree based on recent polymorphism, the haplogroup tree containing only the haplogroups present in the data set and the consensus tree based on both recent and ancient polymorphisms.

Building the recent polymorphism phylogeny

The recent polymorphism phylogeny is built using the neighbor-joining (NJ) algorithm (Saitou 1987). This popular tree construction method follows an agglomerative construction scheme: it starts with a star tree and, until no more than three taxa remain, selects a pair of taxa, creates a node representing this cluster and replaces both taxa by this node (Gascuel 1997; Yang and Rannala 2012). The criterion for selecting which taxa to aggregate is to minimize the sum of branch lengths of the tree, thus following the minimum-evolution principle (first described by Kidd and Sgaramella-Zonta 1971; Gascuel 1997).

NJ algorithm can be applied as long as a genetic distance can be assigned: it is a distance-based method, meaning that the tree is built upon a distance matrix (Yang and Rannala 2012). The distance matrix is computed differently according to the type of input data. If aligned sequences are provided, the distance between individuals is computed by means of a substitution model (Yang and Rannala 2012). We chose the Kimura 2-parameter model (K80; Kimura 1980) that assumes equal bases frequencies but different transversion and transition rates. In our program, we used a distance function already existing in R (ape package (Paradis et al. 2004)) and implemented it in Python. In both cases, a matrix containing pairwise distances is returned. On the other hand, if microsatellites data are provided as input, the Bruvo's distance between individuals is calculated (Bruvo et al. 2004). This method allows to compute genotype distance while taking into account mutation processes (Bruvo et al. 2004). The distance between two alleles is defined as follows:

$$d = 1 - 2^{-|x|} \tag{1}$$

where x is the number of repeat differences (length of the repeated sequence divided by the length - i.e. number of nucleotides - of the microsatellite). The length of the NRY markers was retrieved from an online database (www.cstl.nist.gov/biotech/strbase/ystr_fact.htm, accessed 20 October 2015). If the data set includes markers that are not present in this list, their default length is set to 1. The distance between two individuals is then the mean allelic distance, i.e. the sum of the distance for all alleles divided by the number of alleles (Bruvo et al. 2004). The Bruvo's distance was developed for the comparison of genetic data among individuals with different ploidy level, including polyploids. In this case, permutations should be performed to take into account all possible allele combinations, and the one leading to the smallest sum is finally retained (Bruvo et al. 2004). As the NRY is at haploid level, there is no need for such permutations. Therefore, we could implement this calculation in a simpler manner than other libraries (e.g. the polysat R package (Clark and Jasieniuk 2011)) do.

This flexible distance calculation according to the data type is one of the advantages of the NJ algorithm. Another convenience of this algorithm is its computational speed. Lastly, it is consistent even if small noise perturbs the distance matrix (Gascuel and Steel 2006) and has been proven to perform well in various evolutionary histories (Kalinowski 2009).

Retrieving and pruning the ancient polymorphism phylogeny

After having built the recent polymorphism genealogy, we tackle now the construction of the ancient polymorphism phylogeny. This part of the implementation is more concise because the haplogroup tree is supplied as input. So, two main processes have to be performed. First, the haplogroup tree is pruned: the haplogroups that are not represented in the data set (also inputed) are removed. After that, haplogroups that characterize one or more persons but that are at node positions in the ancient polymorphism phylogeny are shifted to the tip of the tree (terminal positions). This is consistent with the fact that individuals sampled for a phylogenetic analysis are contemporary. The phylogeny thus constructed defines lineages characterized by basal mutations. At this point, a partial haplogroup tree that mirrors deep ancestry among individuals is hence created.

Joining recent and ancient polymorphism phylogenies into a consensus tree

In the previous stages, different genealogies carrying distinct information have been built. In fact, recent polymorphism cannot be used to infer deep ancestry, while ancient polymorphism is not sufficiently variable for resolving recent genealogy. These shortcomings are addressed at this point of the program. Indeed, during this step, the topology of the haplogroup is kept and determines the oldest part of tree. Then, a recent polymorphism tree is built for each subset of individuals that share the same haplogroup, following the distance calculation and NJ algorithm detailed above. An individual with an adjacent haplogroup serves as the outgroup to root this tree, that is lastly joined at the corresponding tip of the haplogroup genealogy. Thus, the different parts of the program lead at the end to a consensus tree that consists in a better estimation of the actual genetic structure of the population.

Figure informations

Figure 1: mitochondrion and Y-chromosome pictures adapted from https://www.drlam.com/images/Diseased_Cell_adrenal_fatigue_5337_1.png and https://www.icr.org/i/wide/y_chromosome_wide.jpg respectively.

References

Bruvo, R., N. K. Michiels, T. G. D'Souza, and H. Schulenburg (2004). "A simple method for the calculation of microsatellite genotype distances irrespective of ploidy level". *Molecular Ecology* 13.7, pp. 2101–2106.

Chapman, B. and J. Chang (2000). "Biopython: Python tools for computational biology". *ACM SIGBIO Newsletter* 20, pp. 15–19.

Charif, D. and J.R. Lobry (2007). "SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis." *Structural approaches to sequence evolution: Molecules, networks, populations.* Ed. by U. Bastolla, M. Porto, H.E. Roman, and M. Vendruscolo. Biological and Medical Physics, Biomedical Engineering. New York: Springer Verlag, pp. 207–232.

Clark, L. V. and M. Jasieniuk (2011). "polysat: An R package for polyploid microsatellite analysis". *Molecular Ecology Resources* 11.3, pp. 562–566.

Gascuel, O. (1997). "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data." *Molecular Biology and Evolution* 14.7, pp. 685–695.

Gascuel, O. and M. Steel (2006). "Neighbor-Joining Revealed". *Molecular Biology and Evolution* 23.11, pp. 1997–2000.

Kalinowski, S. T. (2009). "How well do evolutionary trees describe genetic relationships among populations?" *Heredity* 102.5, pp. 506–513.

Kidd, K. K. and L. A. Sgaramella-Zonta (1971). "Phylogenetic analysis: concepts and methods." *American Journal of Human Genetics* 23.3, pp. 235–252.

Kimura, M. (1980). "A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences". English. *Journal of Molecular Evolution* 16.2, pp. 111–120.

Paradis, E., J. Claude, and K. Strimmer (2004). "APE: analyses of phylogenetics and evolution in R language". *Bioinformatics* 20, pp. 289–290.

Saitou N.and Nei, M. (1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees". *Molecular Biology and Evolution* 4.4, pp. 406–425.

Yang, Z. and B. Rannala (2012). “Molecular phylogenetics: principles and practice.” *Nature Reviews Genetics* 13.5, pp. 303–314.