Introduction

HaploGrouper is a software that can be to classify haplotypes into haplogroups on the basis of a known phylogenetic tree.This is a command line tool written in python and has the following package dependencies.
numpy, version 1.16.5 or higher
gzip

Downloading and running Haplgrouper

```
git clone https://gitlab.com/bio_anth_decode/haploGrouper.git
cd haploGrouper/

python hGrpr2.py -h
usage: hGrpr2.py [-h] -v VCFFILE -o OUTFILE -l HGRPLOCUSFILE -t
HGRPTREEFILE
                 [-i IDLISTFILE] [-r REGIONS] [-c CHROM] [-f
REFERENCEFASTA]
                 [-m] [-w WEIGHTFILE] [-x VERBOSEFILE]

Determine haplogroup for list of individuals based on VCF file and
haplogroup
tree. ----

optional arguments:
  -h, --help            show this help message and exit
  -v VCFFILE, --vcfFile VCFFILE
                        path of vcfFile to be converted (can be
gzipped vcf
                        file)
  -o OUTFILE, --outFile OUTFILE
                        path of output file
  -l HGRPLOCUSFILE, --hGrpLocusFile HGRPLOCUSFILE
                        path of file information about loci used to
assign
                        individuals to haplogroups and the branches
in the
                        haplogroup tree
  -t HGRPTREEFILE, --hGrpTreeFile HGRPTREEFILE
                        path of file information all branches in the
                        haplogroup tree
  -i IDLISTFILE, --IDListFile IDLISTFILE
                        path of file with subset of IDs from VCF
file that are
                        to be used for haplogroup assignment
  -r REGIONS, --regions REGIONS
                        Only base haplogroup assignment on positions
from the
                        specified regions (format: startPos-
stopPos,startPos-
                        stopPos) Multiple regions can be specified
  -c CHROM, --chrom CHROM
                        Only process loci from vcf file with this
chromosome
```

```
                        name. This must be used when the vcf file
contains
                        loci from multiple chromosomes.
  -f REFERENCEFASTA, --referenceFasta REFERENCEFASTA
                        Read reference sequence from fasta file.
This is
                        useful when the vcfFile is based on full
sequence
                        data, but only reports polymorphic positions
or
                        differences from the reference sequence. In
such
                        cases, many phylogenetically informative
positions
                        would be ignored. When the reference
sequence is
                        provided, positions not reported in the
vcfFile are
                        assumed to have the reference state for all
                        individuals in the file.
  -m, --mismatchLoc     Report genotypes of mismatching loci in
outFile
  -w WEIGHTFILE, --weightFile WEIGHTFILE
                        Give mutations differing weights read from a
separate
                        file. This is only useful for loci with a
high rate of
                        recurrent mutations - like the mtDNA control
region.
                        The file should be tab-delimited with four
columns:
                        pos ancAl derAl weight. Mutations not
included in the
                        file default to a weight of 1
  -x VERBOSEFILE, --verboseFile VERBOSEFILE
                        path of file for full matrix of scores for
each node
                        in the tree (lines) for each individual
(columns)
```

Example 1: Running on a single individual for determining the mitochondrial haplogroup

```
echo "HG00096" > docs/HG00096.txt

python hGrpr2.py -v data/ALL.chrMT.phase3_callmom-
v0_4.20130502.genotypes.vcf -t data/mt_phyloTree_b17_Tree2.txt -l
data/mt_phyloTree_b17_Mutation.txt -f data/rCRS.fasta -i
docs/HG00096.txt -o docs/HG00096_mt_hg.txt -x
docs/HG00096_mt_allScores.txt
```

The haplogroup label written out for the above example

ID	Haplogroup	netScore	matchScore	mismatchScore	mismatchLoci	backMutCnt	pruning	allMaxNetScore
HG00096	H16a1	45	48	3		3		H16a1[48-3]

If the user wishes to examine scores of all nodes , then this file can be examined `docs/HG00096_mt_allScores.txt`

```
sort -k 3nr docs/HG00096_allScores.txt |head
```

hGrp	NodeDepth	HG00096
A1	1	38-13
A1a1	1	38-16
A1a	2	38-15
A2	6	38-18
A2_C64T	1	38-19
A2_C64T_G153A	1	39-19
A2_C64T_G16129A	1	38-20
A2_C64T_T16111C	1	39-19
A2_C64T_T16189C	1	38-20
A2a1	1	38-20
A2a3	1	38-20

Example 2: Running on a single individual for determining the mitochondrial haplogroup using the weight option.

```
python hGrpr2.py -v data/ALL.chrMT.phase3_callmom-
v0_4.20130502.genotypes.vcf -t data/mt_phyloTree_b17_Tree2.txt -l
data/mt_phyloTree_b17_Mutation.txt -i docs/HG00096.txt -o
docs/HG00096_hg_wt.txt -x docs/HG00096_allScores.txt -w
data/mt_phyloTree_b17_mutWeights.txt
```

The haplogroup label written out for the above example in the file `docs/HG00096_hg_wt.txt` is given below

ID	Haplogroup	netScore	matchScore	mismatchScore	mismatchLoci	backMutCnt	pruning	allMaxNetScore
HG00096	H16a1	460.1	23.1		3			H16a1[460.1-23.1]

If a partial VCF file is used, then the region parameter must be specified, else haplogroup assigment will be of the reference sequence.

Example 3: Running on a single individual for determining the Y chromosome haplogroup

```
python hGrpr2.py -v
data/ALL.chrY.phase3_integrated_v2a.20130502.genotypes.vcf -i
docs/HG00096.txt -t data/chrY_hGrpTree_isogg2016.txt -l
data/chrY_locusFile_b37_isogg2016.txt -o docs/HG00096_y_hg.txt -x
docs/HG00096_y_hg_allScores.txt
```

The haplogroup label written out for the above example

ID	Haplogroup	Net Score	match Score	Mismatch Score	Mismatch Loci	backMutCnt	pruning	allMaxNetScore
HG00096	R1b1a2a1a2c1k1	698	698	0		0		R1b1a2a1a2c1k1[698-0]

Example 4: Running on a single individual for determining the Y chromosome haplogroup, using the 2019 ISOGG file

```
python hGrpr2.py -v
data/ALL.chrY.phase3_integrated_v2a.20130502.genotypes.vcf -t
data/treeFileNEW_isogg2019.txt -l data/snpFile_b37_isogg2019.txt -i
docs/HG00096.txt -o docs/HG00096_y_hg_ISOGG2019.txt -x
docs/HG00096_y_hg_allScores_ISOGG2019.txt
```

The haplogroup label written out for the above example

ID	Haplogroup	netScore	match Score	mismatch Score	mismatchLoci	backMutCnt	pruning	allMaxNetScore
HG00096	R1b1a1b1a1a2c1a1f1	**952**	**952**	0		0		R1b1a1b1a1a2c1a1f1[952-0]

Example 5: Running on a single individual for determining the Y chromosome haplogroup, using the 2019 ISOGG file, when builds are misspecified

```
python hGrpr2.py -v
data/ALL.chrY.phase3_integrated_v2a.20130502.genotypes.vcf -t
data/treeFileNEW_isogg2019.txt -l data/snpFile_b38_isogg2019.txt -o
docs/HG00096_ISOGG2019_y_hg.txt -i docs/HG00096.txt
```

```
############ Running haplogrouper ############
vcf file: data/ALL.chrY.phase3_integrated_v2a.20130502.genotypes.vcf
Haplogroup assignment will be based on
treeFile: data/treeFileNEW_isogg2019.txt
locusFile: data/snpFile_b38_isogg2019.txt
Results will be written to file: docs/HG00096_ISOGG2019_y_hg.txt

## Reading haplogroup tree and locus files
Node YRoot in line 0 has parent  that is not present in the tree. This should only happen for root node
position 19449005..19450128 is not numeric and will be ignored
position 12812702..12812703 is not numeric and will be ignored
nodeName #REF! not found in path file. Should not happen!
position 19995095..19995096 is not numeric and will be ignored
position 12925842..12925843 is not numeric and will be ignored
position 19764155..19764156 is not numeric and will be ignored
nodeName #REF! not found in path file. Should not happen!
9384 of 9402 nodes have at least one mutation and will be used (18 have none and will be ignored)
Read 71399 mutations at 70004 positions tagging 9384 haplogroup labels from tree and snp files

## Scoring based on vcfFile
VCF file is not gzipped
```

Found 1 IDs that overlap between 1 from IDListFile and 1233 from VCF file
Processing GTs from vcfFile[. for every 1000 loci].....Done in 2.72 seconds.
Used GTs from 248 of 70004 positions listed in hGrpLocusFile (62041 in the vcfFile)

Making assignments based on 226 of 9384 haplogroup labels that were encountered for scoring
Finished in 2.728850 seconds

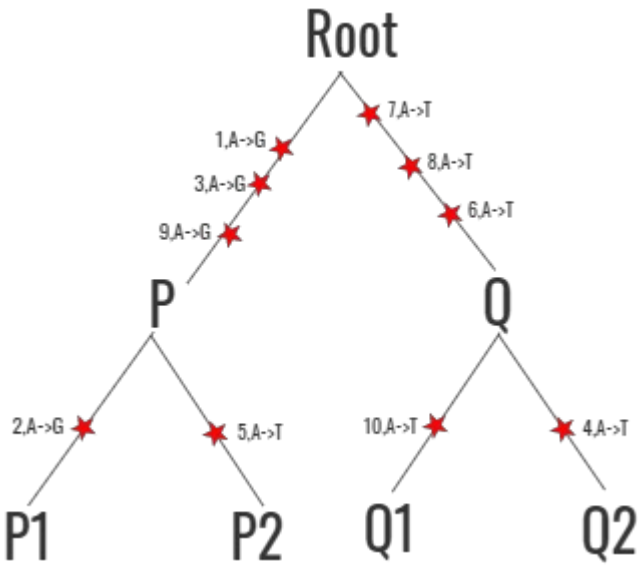
The haplogroup label written out for the above example

ID	Haplogroup	netScore	matchScore	mismatchScore	mismatchLoci	backMutCnt	pruning	allMaxNetScore
HG00096	S2	4	4	0		0		S2[4-0]

Creating custom tree and variant files:

The tree file used in our software consists of two columns, the first column contains the node name of the tree and the second column the parent node. The variant files consists of five columns , where columns one to five contains the variant name, physical position, allele supporting the node ,name of the node (haplogroup label) and the ancestral allele.

Let's consider a simple example



The tree file for this example

Root	
P	Root
P1	P
P2	P
Q	Root
Q1	Q
Q2	Q

The variant file for this example looks like this

SNPName	Position	DerivedAllele	NodeName	AncestralAllele
Rs1	1	G	P	A
Rs2	2	G	P1	A
Rs3	3	G	P	A
Rs4	4	T	Q2	A
Rs5	5	T	P2	A
Rs6	6	T	Q	A
Rs7	7	T	Q	A
Rs8	8	T	Q	A
Rs9	9	G	P	A
Rs10	10	T	Q1	A