**Instruction manual for Perl programs for construction and analysis of haplotypes of frequent genetic variants.**

**Human Chromosomes and 1092 Individuals:**

The 1000 Genome Phase I database contains SNP data for all 22 human autosomes and the ‘X’ chromosome for each of the 1092 individuals from 14 different populations belonging to four different continents. Version 4.1 of this dataset contains a variety of genetic variants (GVs) - a total of 38.2M SNPs, 3.9M short indels and 14K deletions for all the human chromosomes. Since the majority of the GVs are SNPs, here we will use the word SNP for all GVs.

**The 1000Genomes data files in INTRON2:**

The 1000Genomes data files are stored in a specific directory of the supercomputer intron2 workstation (refer figure 1 below).



**Figure 1: screenshot of directory with 1000Genomes data files in the supercomputer intron2 workstation**

**STEP I: Extraction of SNPs and building the haplotypes:**

The perl script ***Haplo\_find.pl*** opens and processes the 1000 Genomes .vcf.gz file for the desired chromosome. This program filters the SNPs with desired frequency and processes them into haplotypes. The user chooses the parameters (such as the chromosome of choice, desired frequency of SNPs, number of SNPs in the haplotype and a threshold count for defining the Common haplotypes) according to preferences in the arguments in the command line. For example, let us consider the following screenshot of a command line for executing ***Haplo\_find.pl****:*



**Figure 2: screenshot of command line for execution of Haplo\_Find.pl**

The command line in the figure 2 above computes the haplotypes for Chromosome **1 (argument 1)**, with SNPs having frequency >**0.25** **(argument 2)** for minor alleles (MAF). The program constructs haplotypes with **50 (argument 3)** such adjacent frequent SNPs. Common haplotypes will be those which occur >=**100 (argument 4)** times in the populations (1092 individuals).

While processing the .vcf.gz file for the intended chromosome, the script ***Haplo\_find.pl*** considers the first SNP coordinate in the chromosome as the beginning of the first **SEGMENT**. In the 1000Genomes dataset, the SNP coordinate is in the second column under the header ‘POS’ and the SNP ID is in the 3rd column with ‘ID’ header (see screenshots below). The program starts scanning the file for SNPs with user specified frequency (>0.25 MAF in the given example). The frequency is denoted as AF and provided in the INFO column (column #8, see figure 3 below).

 

**Figure 3: screenshots of 1000Genomes vcf.gz file format**

When the program finds the first SNP with the desired frequency, it notes the coordinate and keeps looking for the desired SNPs until it finds the 50th such SNP. When the 50th SNP is found, the coordinate of the 50th SNP is marked. The distance between the 1st and the 50th SNP is defined as the **HAPLOTYPE LENGTH**. If the HAPLOTYPE LENGTH is <= 500,000, the next segment starts from the coordinate of the 1st SNP + 500000. If the HAPLOTYPE LENGTH is > 500,000, the starting coordinate of the next segment becomes (the 50th SNP coordinate + 1).

The 50 lines corresponding to the 50 filtered SNPs, containing the general information and genotype information of the 1092 individuals are saved in output files named **ALL\_STATS\_Cx-y** (where x denotes the Chromosome number and y denotes the segment number). The genotype information of the 1092 individuals are then extracted by the program.

Genotype annotations for an individual homozygous with mutant genetic variants is annotated 1|1 in the VCF file. Homozygous presence of reference alleles in an individual is annotated 0|0. Genotype of heterozygous individual is annotated 0|1 or 1|0. These genotype information is available from the column respective to each individual identified by a unique identifier (HGxxxxxx or NAxxxxxx) in the header line. The figure 4 below shows these genotype information and Individual Identifiers from one of the .vcf.gz files. Since the 1000Genomes data is phased, we have denoted the first genotype as M (maternal) and the second as P (paternal) for ease of understanding, although practically it is not possible to identify the maternal and paternal chromosome with certainty from this data.



**Figure 4: screenshot of genotype annotation in one of the 1000Genomes data files**

The program Haplo\_Find.pl extracts and stores these genotype information for each of the 1092 individuals for the 50 SNPs of interest (MAF >0.25) and constructs the two haplotypes (maternal and paternal) for each of the 1092 individuals. Practically, each haplotype is a string consisting of 50 single digits- 0s and 1s.



**Figure 5: screenshot showing haplotypes in one of the output files named ALL\_STATS\_Cx-y**

From the .vcf file, the program also extracts the individual identifier. The population each individual comes from is available from the 1000Genomes website. These two information are then combined together by the program to create a unique identifier along with M (for the maternal chromosome) and P (for the paternal chromosome). Thus, for example, in the figure 5 above, two chromosomal strands for the individual highlighted are denoted as GBR\_HG00096\_M and GBR\_HG00096\_P where GBR is the population and HG00096 is the individual identifier. M and P stands for maternal and paternal respectively.

The haplotypes are stored along with the respective identifiers in the output files named **ALL\_STATS\_Cx-y** (where x denotes the Chromosome number and y denotes the segment number).

Finally, the algorithm compares the 2184 haplotypes (strings of 0s and 1s as shown in the figure above, 2 \* 1092 = 2184) with each other and counts the occurrences of the haplotypes. The haplotypes that occur more than the user specified threshold (in the example here >=100) are defined as Major haplotypes. These common haplotypes are listed at the bottom of the output file **ALL\_STATS\_Cx-y** (see figure 6 below).



**Figure 6: screenshot showing Major haplotypes in one of the output files named ALL\_STATS\_Cx-y**

These processes continue for the entire length of the chromosome and the resulting segments, their haplotypes and other information are saved in automatically created output files for each respective segment.

This STEP requires about 1.5 hr to finish computation for Chromosome 2 which is one of the largest chromosomes and have the highest number of SNPs with frequency >0.25 (233,023). Therefore, for each of the other chromosomes, complete execution of this program requires less than 1.5 hr.

Upon execution, this program automatically generates the following output files:

Outputs:

1. **ALL\_STATS\_Cx-y** for all segments. For example:

**% vi ALL\_STATS\_C1-143**

This output file was created on Sun Mar 13 00:08:06 2016

CHR\_1 Segment\_143 Hap Length= 49259

Total no. of SNPs in this segment= 706

SNPs with freq > 0.25 & < 0.75 are:

1 71320836 rs34747258 GA G 442 PASS AA=GA;AC=605;AF=0.28;AFR\_AF=0.22;AMR\_AF=0.30;AN=2184;ASN\_AF=0.20;AVGPOST=0.9943;ERATE=0.0006;EUR\_AF=0.36;LDAF=0.2768;RSQ=0.98

1 71325044 rs1327460 T C 100 PASS ERATE=0.0004;AVGPOST=0.9985;AA=C;AC=1376;AN=2184;VT=SNP;LDAF=0.6295;THETA=0.0006;SNPSOURCE=LOWCOV;RSQ=0.9975;AF=0.63;ASN\_AF=0

1. **STAT\_Chr\_1-SNP\_50-HAP\_100**

**% vi STAT\_Chr\_1-SNP\_50-HAP\_100**

This output was generated on Sat Mar 12 23:52:29 2016

#of\_Major\_HAPs Segment\_Occurrence

0 70

1 47

2 53

3 90

4 96

5 67

6 20

7 11

8 2

1. **HAP\_Length\_C\_x-SNP\_50-HAP\_100**

**% vi HAP\_Length\_C\_1-SNP\_50-HAP\_100**

This output was generated on Sat Mar 12 23:52:29 2016

Seg\_No Hap\_L

1 771170

2 43840

3 19193

4 52649

5 49228

6 36368

7 23733

1. **No\_of\_Major\_HAP\_C\_x-SNP\_50-HAP\_100**

**% vi No\_of\_Major\_HAP\_C\_1-SNP\_50-HAP\_100**

This output was generated on Sat Mar 12 23:52:29 2016

Seg\_No No\_of\_Maj\_Hap

1 0

2 0

3 6

4 3

5 1

6 4

7 4

1. **TABLE\_C\_x-SNP\_50-HAP\_100**

**% vi TABLE\_C\_1-SNP\_50-HAP\_100**

Seg\_No Hap\_L No\_Major\_Hap Top\_Hap\_Ct 2nd\_Hap\_Ct 3rd\_Hap\_Ct

1 771170 0 11 11 10

2 43840 0 54 53 30

3 19193 6 548 227 198

4 52649 3 299 140 133

5 49228 1 133 97 55

6 36368 4 313 176 134

1. **TOP\_HAP\_CT\_C\_x-SNP\_50-HAP\_100**

Lists the integer value of **[Top Hap count in the segment/50].** The top hap in Seg 4 occurs 274 times. So for Seg 4, this file will have a value of (274/5) = 5.

**STEP II:YIN, YANG and MOSAIC Haplotypes**

1. **Grouping of nearly identical haplotypes into ‘Haplotype Groups’ and identifying the Common Haplotypes**

This step uses a system call (**system\_for\_Yin\_Yang.pl)** to run the **Hap\_Group\_generator.pl** program for all the segments created for each chromosome in STEP I.

The command line for execution of this step is as follows:

**% perl system\_for\_Yin\_Yang.pl**

The *Perl* script **Hap\_Group\_generator.pl** uses the filesnamed **ALL\_STATS\_Cx-y** generated in step 1 as inputs for each segment. The program scans the input file for each segment and then compares the 2184 haplotypes (from the 1092 individuals) and identifies haplotypes which have <=2 differences between them. Such haplotypes are defined as a ‘**haplotype group’.** Such a haplotype group is represented by the haplotype that has the maximum occurrence within that group. The list of all members of a haplotype group is saved in the output file named **Hap\_Grouping\_New\_x** (where x denotes the segment number) along with their respective occurrences in decreasing order (see figure 7 below). Note that the differences between the representative haplotype and other haplotypes in group 1 have been underscored in red in figure 7.



**Figure 7: screenshot showing grouping of haplotypes in one of the output files named Hap\_Grouping\_New\_x**

The combined occurrence of all group members is considered as the group occurrence and is considered to define effective Common haplotypes. Thus, a haplotype which is not listed as a Major haplotype in the STEP I output file **ALL\_STATS\_Cx-y** (due to occurrence <100 times) could feature in the STEP II output file **Hap\_Grouping\_New\_x** by virtue of additional occurrences of similar haplotypes (haplotype group members with 1 or 2 differences) leading to a combined total occurrence of >=100. For example, let us consider the following figure 8. In this case, the 1st member of Group 5 occurs 71 times in the segment. Therefore, per our definition of Major haplotypes (occurrence >= 100 times), this haplotype is **NOT** a Major haplotype. So, the respective **ALL\_STATS\_Cx-y** file will not have this haplotype listed as a common haplotype. But as we find the other members of Group 5, their occurrences add up and increases the Group occurrence to 105. Therefore, Group 5 qualifies as a common haplotype. Since Group 5 will be represented by its 1st member (with highest occurrence = 71), the first member now is considered a common haplotype.



**Figure 8: screenshot showing grouping of haplotypes**

The program also translates the population information of the individuals into continent information and computes the continental distribution of the Common haplotype groups. In the figure above, AF denotes the African continent while AS and EU denotes Asia and Europe respectively. AMR denotes the America and ASW denotes ‘America South West’. If a haplotype group occurs predominantly (>=80%) times in one of these continents, it is defined as a ‘continent specific’ haplotype (for example, Asia specific haplotype and so on). Such continent specificity is noted as ‘AS\_SP’ followed by the percentage of occurrence at the end of the continent information line (for example, **AS\_SP 90%**). If predominant occurrence is not observed, the haplotype is considered to have ‘no continent specificity’ and this is noted as **‘no\_sp’** at the end of the continent information line (see figure above). All the above information are saved in automatically generated output files named **Hap\_Grouping\_New\_x,** where x denotes the segment number.

The common haplotypes are then listed and saved into a second set of automatically generated output files named **CORE\_HAPS\_New\_x**, where x denotes the segment number. The continental distribution and the ancestral and derived allele percentage of the common haplotypes are also saved in the same output files.

R stands for ‘reference allele’ while M denotes mutant/alternative allele. The alleles for which ancestral information is not available information are represented by ‘x’ in the output file (for example, see the figures 7 and 8 above). Using the ancestral allele information available from the vcf.gz file, the algorithm deduces the ancestral state of each allele in each common haplotype. In the ancestral state, ‘A’ denotes ancestral allele and ‘D’ denotes derived alleles. The unknown ‘x’ positions are kept as ‘x’. Then, the program calculates the percentage of ancestral and derived alleles in each common haplotype. The loci denoted with ‘x’ are not considered while calculating the ancestral and derived allele percentage. These information are also stored into the **CORE\_HAPS\_New\_x**, output files. Below is a screenshot of a portion of one such file.



**Figure 9: screenshot of one prototype of output files named CORE\_HAPS\_New\_x**

In addition to these information, the **CORE\_HAPS\_New\_x** files also contain relevant general information about the segment (see the top of the file in figure 9).

1. **Identifying YIN, YANG and MOSAIC haplotypes**

Next, the algorithm compares the Common haplotype groups with each other and counts differences. If two haplotype groups are found to have >=47 differences between them, they are defined as a pair of YIN and YANG haplotypes. The common haplotype groups other than the YIN and YANG are considered as mosaics since these carry similarities to both YIN and YANG. For better clarity, a MOSAIC haplotype can be perceived as consisting multiple pieces from the YIN and YANG haplotypes. The number of differences between the common haplotype groups and the YIN, YANG, and MOSAIC haplotypes are listed in the files named **CORE\_HAPS\_New\_x.** This is illustrated in the following figure 10. We can see that the # of differences between Group1 and Group3 is 49. Therefore, Group1 and Group3 are a YIN-YANG pair. Since the differences between all other common haplotype groups is less than 47, we have only one YIN-YANG pair in this segment. The program also computes the number of such pieces in a mosaic group by comparing it with the YIN and YANG at the same time.



**Figure 10: screenshot showing a different portion of the output files named CORE\_HAPS\_New\_x**

If a segment has more than one YIN-YANG pair, the program identifies all of them and also lists mosaics for each YIN-YANG pair separately. Number of pieces in a mosaic group is also computed in each case by comparing with the respective YIN and YANG pair. The following figure 11 illustrates this. The segment has two YIN-YANG pairs since the differences between Group1 and Group 4 is 50 while the differences between Group2 and Group 4 is 47.



**Figure 11: screenshot showing portion of the output files named CORE\_HAPS\_New\_x having >1 Mosaic**

In addition to the **Hap\_Grouping\_New\_x** and **CORE\_HAPS\_New\_x files,** other output files generated at the end of STEP II are **STAT\_FOR\_Mos\_New\_x** and **STAT\_FOR\_Yin\_Yang\_New\_x, where x denotes the Chromosome number. These** output files contains all the statistics for Mosaic haplotypes and YIN-YANG haplotypes respectively. Another short script **combine\_stats\_YY\_with\_Mos.pl** is used to combine statistics for YIN-YANG and Mosaics together and generate the output file **Combined\_STATS\_YY\_Mos\_x**, x denoting the chromosome number.

22 different sets of 3 output files for each of the 22 autosomes are generated in this step and those files are again analyzed to calculate the statistical summaries in the **STEP IV**.

To view **STAT\_FOR\_Yin\_Yang\_New\_1** see **Supplementary\_File\_S4.xlsx.**

To view **STAT\_FOR\_Mos\_New\_1** see **Supplementary\_File\_S5.xlsx.**

To view **Combined\_STATS\_YY\_Mos\_1** (for chromosome 1) see **Supplementary\_File\_S6.xlsx.**

STEP II requires 30 minutes to complete all computations and to generate outputs for chromosome 2. Thus, it takes less than 30 minutes for each of the remaining chromosomes.

**STEP III: Analyzing the Mosaic haplotypes and generating statistics for them**

This step uses a combination of 2 perl scripts to generate different statistics for the Mosaic haplotypes. One of those scripts is a system call which is used to execute the primary script for all relevant files. The script **system\_for\_ALL\_MOSAIC.pl** is used to execute the primary script **Mosaic\_STAT\_generator\_ALL\_MOSAIC\_GRs.pl** with the following command line:

**% perl system\_for\_ALL\_MOSAIC.pl**

The primary script **Mosaic\_STAT\_generator\_ALL\_MOSAIC\_GRs.pl** uses the **CORE\_HAPS\_New\_x** files generated at the end of STEP II as the inputs and performs different statistical analysis of all the Mosaic haplotypes from all segments of a chromosome. The output generated in this step is **STAT\_FOR\_Mosaics\_ALL\_GROUPs\_CHR\_x** where x denotes the chromosome number. Following is a screenshot of one such output file



**Figure 12: screenshot of a prototype of the output files named STAT\_FOR\_Mosaics\_ALL\_GROUPs\_CHR\_x**

22 different output files for 22 chromosomes are generated in this step and those files are again analyzed to calculate the statistical summaries in the next step.

Similarly, two other system call-Perl script combinations are used to generate different statistics**.**

1. System call **system\_for\_MOST\_ANCESTRAL\_MOSAIC.pl** is used along with **Mosaic\_STAT\_generator\_MOST\_ANCESTRAL\_MOSAIC.pl.** This alsouses the **CORE\_HAPS\_New\_x** files generated at the end of STEP II as the inputs and performs statistical analysis for the Mosaic haplotypes with most ancestral alleles in each segment. The output generated in this step is **STAT\_FOR\_Mosaics\_MOST\_ANCESTRAL\_CHR\_x** which has a similar structure as the **STAT\_FOR\_Mosaics\_ALL\_GROUPs\_CHR\_x** file shown above (with x being the chromosome number).
2. System call **system\_for\_TOP\_PIECES\_MOSAIC.pl** is used along with **Mosaic\_STAT\_generator\_TOP\_PIECES\_MOSAIC.pl. T**his alsouses the **CORE\_HAPS\_New\_x** files generated at the end of STEP II as the inputs and performs different statistical analysis for the Mosaic haplotypes with maximum number of pieces in each segment. The output generated in this step is **STAT\_FOR\_mosaics\_TOP\_PIECES\_CHR\_x,** with x being the chromosome number. This output file also has a similar structure as the **STAT\_FOR\_Mosaics\_ALL\_GROUPs\_CHR\_x** file shown above.

22 different output files for 22 chromosomes are generated in both cases and those files are again analyzed to calculate the statistical summaries in the next step.

It takes 20 minutes to complete all STEP III computations and generate outputs for chromosome 2.

**STEP IV**: **Generating statistical summary for YIN, YANG and MOSAIC haplotypes.**

This step uses three different Perl scripts to generate summary statistics for Yin, Yang and Mosaic haplotypes.

**Yin\_Yang\_STAT\_explorer.pl** uses **STAT\_FOR\_Yin\_Yang\_CHR\_x** files(generated in STEP II) as inputs and computes the summary statistics for Yin and Yang haplotypes, which is saved in the output file **Analysis\_YIN\_YANG\_Freq\_0.25.**

This output file is transferred into the PC/Mac environment from the Linux environment in MS Excel format by **sftp** and further analyzed with MS Excel.

**Mosaic\_STAT\_explorer.pl** uses **STAT\_FOR\_Mosaics\_ALL\_GROUPs\_CHR\_x** files(generated in STEP III) as inputs and computes the summary statistics for all Mosaic haplotypes, which is saved in the output file **Analysis\_Mosaics\_All\_Gr\_Freq\_0.25.** This output file is transferred into the PC/Mac environment from the Linux environment in MS Excel format by **sftp** and further analyzed with MS Excel.

Similarly, **Mosaic\_STAT\_explorer\_2.pl** is used to generatesummary statistics for Mosaic haplotypes with most ancestral alleles in each segment using the files **STAT\_FOR\_Mosaics\_MOST\_ANCESTRAL\_CHR\_x.**

**Mosaic\_STAT\_explorer\_3.pl** is usedtogeneratesummary statistics for Mosaic haplotypes with maximum number of pieces in each segment using the files **STAT\_FOR\_mosaics\_TOP\_PIECES\_CHR\_x** as inputs.

The output files in these lasttwo cases are named **Analysis\_Mosaics\_MOST\_ANCESTRAL\_Freq\_0.25** and **Analysis\_Mosaics\_TOP\_PIECES\_Freq\_0.25** respectively**.**

These output files are transferred into the PC/Mac environment from the Linux environment in MS Excel format by **sftp** and further analyzed with MS Excel.

All STEP IV computations are completed and all output files are generated within 25-30 minutes for chromosome 2.

**NOTE: For calculating different statistics for our study, we have used Mosaic\_STAT\_explorer.pl and Mosaic\_STAT\_explorer\_3.pl.**

**STEP V: Building the Denisovan diplotypes and comparing them with the Human Common Haplotypes**

The vcf.gz files for the 22 Denisovan autosomes are stored in the following directory in our intron2 supercomputer workstation.



**Figure 13: screenshot showing Denisovan vcf.gz files in intron2 supercomputer workstation directory**

In this step, a Perl script **Deni\_extract\_SNP.pl** is used to extract all the SNPs and their relevant information from each of the 22 Denisovan autosomes. These were saved into 22 different output files named **DENISOVAN\_Pinky\_ALL\_SNPs\_x,** where x denotes the chromosome number.

The Perl script **DENISOVA\_PINKY\_HAPLO\_FIND.pl** is used to find haplotypes for each of the 22 Denisovan autosomes and carry out further statistical analyses.

1. **Identifying human chromosomal segments which contain common haplotypes:**

The algorithm uses **STAT\_FOR\_Mosaics\_ALL\_GROUPs\_CHR\_x** files (generated in STEP III) to identify in a particular chromosome (x denotes the chromosome number) the segments where Yin, Yang and Mosaic haplotypes are present. Thus, the segments that do not have Mosaics are skipped since our goal here is to compare all the human common haplotypes with the Denisovan haplotypes. Note that the algorithm extracts only the segments that have one or more Mosaic haplotypes consisting of >=12 pieces. For example, let us take a look at the following figure 14. Segment 4 has 2 MOSAIC haplotypes with 15 and 21 pieces respectively, while segment 5 has 1 MOSAIC haplotype with 15 pieces. The algorithm selects these segments but skips segments 3, 5 and 9 since the MOSAIC haplotypes in those segments consist of <12 pieces.

****

**Figure 14: identifying human chromosomal segments having CHs**

1. **Selecting human haplotype SNPs**

Next, as a second input file, the algorithm opens the **ALL\_STATS\_Cx-y** files for the respective segments identified above **(**where x is the chromosome number and y is the segment number). Thus, for the scenario shown in the picture above, the program opens the **ALL\_STATS\_C1-4** file (see figure 15 below). From each **ALL\_STATS\_Cx-y** file, the program extracts the 50 SNP ids [denoted as rs12345 for known SNP id or dot (.) for unknown SNP id].



**Figure 15: Extracting SNP ids**

1. **Constructing the Denisovan diplotypes:**

The program then opens the third input file **DENISOVAN\_Pinky\_ALL\_SNPs\_x**, where x is the chromosome number. In the current example, the program will open the **DENISOVAN\_Pinky\_ALL\_SNPs\_1** file. The algorithm scans this third input and searches for the extracted SNPs with a known SNP id (identified above). When a match is found in the Denisovan genome, the algorithm extracts the genotype information for the respective SNP and saves it. Since the Denisovan genome sequence data is unphased, the algorithm translates the genotype information into a diplotype instead of haplotype. Genotype annotations corresponding to those GVs with respect to homozygous with genetic variants (annotated 1/1 in the VCF file) are translated into 2 (1+1), homozygous presence of reference alleles (annotated 0/0) are translated into 0 (0+0) and heterozygous (annotated 0/1 or 1/0) are translated into 1 (0+1 or 1+0). SNP for which no match is found is noted; and an ‘x’ is saved for its position in the diplotype.

The Denisova genome contains some SNP ids in duplicate. To avoid complications, we decided not to include those SNPs in our analysis. Therefore, a ‘U’ is saved in the diplotype for such SNPs.

The Denisova genome also contains incomplete genotype information for some SNPs (1/. Or ./1 or 0/. Or ./0). We also excluded those SNPs from our analysis. The algorithm saves an ‘N’ in the diplotype for each SNP with incomplete genotype information in the Denisova genome. Figure 16 below exemplifies such an instant.



**Figure 16: screenshot showing missing genotype information in one of the Denisovan chromosome data files**

For SNPs which has unknown SNP id (dots) in the human genome are represented in the diplotype with an ‘x’. The SNP positions for which no match is found in the Denisovan genome are also denoted as ‘x’.

The algorithm also computes the ancestral and derived state of each allele in the Denisovan diplotype by processing the ancestral allele information from the **CORE\_HAPS\_New\_x** file for the respective segment.

The Denisovan diplotype and the computed ancestral state is then printed along with the contents of the **CORE\_HAPS\_New\_x** file into the output files named **CORE\_HAPS\_with\_DENI\_Pinky\_x,** where x is the segment number. Figure 17 shows a portion of one such output file.



**Figure 17: screenshot showing a prototype of output files named CORE\_HAPS\_with\_DENI\_Pinky\_x**

1. **Generating comparative statistics for denisovan:**

A system call **system\_for\_DENI\_compar\_STAT\_gen.pl** is used to execute the third Perl script in STEP V, namely **Deni\_STAT\_generator.pl**. This processes all the **CORE\_HAPS\_with\_DENI\_Pinky\_x files** for each chromosome and generates the statistics. The command line used is as follows

**%perl system\_for\_DENI\_compar\_STAT\_gen.pl**

The results are saved in an output file named **STAT\_Comparison\_FOR\_DENI\_CHR\_x** (x denotes the respective chromosome number). Following figure 18 is a screenshot of one such output file:



**Figure 18: screenshot showing a prototype of the output files named STAT\_Comparison\_FOR\_DENI\_CHR\_x**

Finally, the Perl script **DENI\_STAT\_explorer.pl** processes 22 such **STAT\_Comparison\_FOR\_DENI\_CHR\_x** files (one for each chromosome) to compute the summary statistics and saves into the output file named **Analysis\_DENI\_All\_Freq\_0.25.** These output file is transferred into the PC/Mac environment from the Linux environment in MS Excel format by **sftp** and further analyzed with MS Excel.

STEP V requires 90 minutes to complete all computations and generate outputs.

**STEP VI: Identifying the presence of Ancestral haplotypes in the Human population**

Ancestral allele information is denoted by the nucleotides A, C, G and T (see green boxes in the figure 19 below) in the 1000Genomes .vcf.gz files. For some SNPs, the ancestral state is not known. The 1000Genomes database denotes ancestral information for such SNPs by a dot [.] (see yellow boxes in the figure 19 below).



**Figure 19: screenshot showing missing ancestral allele information in one of the 1000Genome .vcf.gz file**

Ancestral allele information for each SNP of interest is extracted in STEP I by the Perl script ***Haplo\_find.pl*** which further processes this information and computes an ancestral haplotype for each segment. This ancestral haplotype is a string of 50 SNPs that represent only the ancestral alleles for that particular segment. This ancestral haplotype is saved in the STEP I output files named **ALL\_STATS\_Cx-y** (where x denotes Chromosome number and y denotes the segment number). The *Perl* program ***Ancestral\_Hap\_Match\_Finder.pl*** extracts the ancestral haplotype for each segment from the respective **ALL\_STATS\_Cx-y** file and finds its matches from the 2184 haplotypes in the 1000Genomes population in the corresponding segment. The algorithm compares only the alleles for which the ancestral state is known. It does not consider the alleles with unknown ancestral state. For example, if the ancestral haplotype of a particular segment contains 15 or more alleles with unknown ancestral state, this program skips that segment. Also, this program allows <=2 mismatches while searching for matches. For example, let us consider a segment that has an ancestral haplotype consisting of 40 alleles with known ancestral state and 10 alleles with unknown ancestral state. In this case, the algorithm looks for matches of the ancestral haplotype for those 40 positions. The haplotype that matches 38, 39 or 40 of those alleles is considered a ‘match’.

The primary script **Ancestral\_Hap\_Match\_Finder.pl** is executed for all segments in a chromosome with the help of the system call **system\_for\_Anc\_Hap\_Match\_finder.pl.** The system call provides the user with the option to select the desired chromosome and total number of segments intended to be considered. The Command line for execution of this script is

**%perl** **system\_for\_Anc\_Hap\_Match\_finder.pl. 1 456**

Here the first argument (1) represents the chromosome number and the second argument (456) represents the total number of segments for chromosome 1 (generated in STEP I).

The first output file generated is **MATCH\_Anc\_Hap\_CHR\_x\_2mm** (x denotes the chromosome number).For example, see figure 20 below.

****

**Figure 20: screenshot showing a portion of the output file MATCH\_Anc\_Hap\_CHR\_1\_2mm**

Another output file **No\_Match\_Anc\_Hap\_CHR\_x\_2mm** (x denotes the chromosome number) lists the segments where no ancestral haplotype match was found.

A third output file **CONTINENT\_STATs\_Anc\_Hap\_All\_CHR\_2mm** is also generatedwhich stores the continental distribution of ancestral haplotype matches for all segments from all the 22 chromosomes. Figure 21 below shows a part of this output file:



**Figure 20: screenshot showing a portion of the output file CONTINENT\_STATs\_Anc\_Hap\_All\_CHR\_2mm**

**List of all Perl scripts used in this study:**

***HaploFind.pl;***

***HapGroupGenerator.pl,***

***MosaicStatGenerator1.pl,***

***MosaicStatGenerator2.pl,***

***MosaicStatGenerator3.pl,***

***YinYangStatExplorer.pl,***

***MosaicStatExplorer.pl,***

***MosaicStatExplorer\_2.pl,***

***MosaicStatExplorer\_3.pl,***

***CombineStatsYY\_Mos.pl,***

***AncestralHapMatchFinder.pl*.**

***Deni\_extract\_SNP.pl***

***DENISOVA\_PINKY\_HAPLO\_FIND.pl***

***Deni\_STAT\_generator.pl***

***Deni\_STAT\_explorer.pl***