SNP calling exercise – Tracing the origin of early medieval walrus ivory using ancient DNA

Background

The Viking Age was a period of great mobility in Scandinavia, culminating in the discovery of the new world (North America), and the well-known colonization of Greenland by the Norse Vikings in the 10th century under leadership of Erik the Red. The Norse lived on Greenland in several large settlements for hundreds of years, until their settlements disappeared without a trace in the 15th century, leaving only abandoned ruins. This mysterious disappearance has fueled many hypotheses about what happened in Greenland since the 18th century, which is when Europeans found out that these colonies had disappeared altogether. *Why did they settle there, what were they doing there, and why did they mysteriously disappear?*

One of the theories that that Greenland was settled for the hunt of walruses. Walrus ivory was a popular material for the manufacture of luxury art objects in during the Viking age and early medieval Europe. But the Atlantic walrus is widely distributed in the Arctic—with populations being found from Siberia, Svalbard, eastern Greenland, western Greenland (where the Norse settled in the Viking age) to Canada. Where did all this medieval walrus ivory really come from? The hunting of walruses for ivory by Norse Greenlanders is testified by archaeological finds of skulls, skull fragments, tusk offcuts, cheek teeth, and carved objects. But was this for local use? Or for export? There exists no empirical evidence on the geographical origin of the ivory imported to European trading centres such as Trondheim, Bergen, Oslo, Dublin, London, Sigtuna, and Schleswig during the chronology of the Norse Greenland settlements.

Your task:

You are a MSc candidate with bioinformatic expertise; you run a DNA project that tries to provide new insights into the questions above. Specifically, can you find any evidence for trade of walrus ivory from Norse Greenland? For this, you have obtained genomic data (ancient DNA data) extracted from 36 medieval walrus artifacts that were found in archaeological excavations in Europe. It is your task to find out if it is possible to genetically identify walrus specimens with a direct link to Greenland. Some *reference specimens* have also been included. These *reference* specimens either come from western Greenland in the western Atlantic:

WLR030 (reference western Greenland)
WLR031 (reference western Greenland)

whereas other reference specimens have been obtained from Svalbard in the eastern Atlantic:

WLR008 (reference Svalbard)
WLR009 (reference Svalbard)
WLR010 (reference Svalbard)
WLR011 (reference Svalbard)
WLR012 (reference Svalbard)

These are reference specimens that have a known location, being retrieved from these locations. Also included is a Pacific walrus specimen: WLRPAC.

This is a different walrus species that does not naturally occur in the Atlantic Ocean and you will use this as an “outgroup”.

All the other specimens come from archaeological excavations in Europe (Trondheim, Bergen, Oslo, Dublin, London, Sigtuna, and Schleswig). But it is clear that walruses do not actually occur there, so they must have come from elsewhere.

Some of the initial bioinformatic work has been done for you. A large number of sequencing reads have been generated from DNA obtained from the ancient Atlantic walrus specimens that were found in various archaeological size in Europe. What you have been given is a number of BAM files (binary alignment files, and their associated index files) that contain read alignments of these ancient walrus reads towards a subset set of the Atlantic walrus reference genome —in this case it is the *mitochondrial genome.* Mitochondria are organelles inside mammalian cells that contain small genomes that segregate separate from nuclear genomes and are maternally inherited (google mitochondrial genome if you want to learn more). **The great advantage of studying mitogenomes is that they are small in size, and therefore easy to work with and computationally fast to analyze.** Most of the basic principles for calling SNPs are similar for these mitogenomes compared to autosomal genomic regions.

Q: Discuss one aspect of mitochondrial SNP calling that differs from autosomal SNP calling. What is this?

Let's organize the data for you to start working with these data: assume you have SOME experience, with using UNIX and the FOX cluster.

Open your terminal/GITBash window and login to:
ssh ec-<username>@fox.educloud.no
(fill in your own username)
Follow instructions with the 1-time code and password.

Then login to one of the compute nodes (int-1, int-2, int-3, or int-4).
ssh int-<x>
Selected the node (e.g. *int-3* if that is least occupied)
Follow instructions with the 1-time code and password.

Once you've logged in to one of the int nodes, make a new directory:

mkdir walrus_SNP_calling

do
cd walrus_SNP_calling
and
pwd

You should see something like:
/fp/homes01/u01/ec-<username>/walrus_SNP_calling

Whereby your username is filled in instead of username.

Now copy the walrus data you for your own personal use to your own directory (notice the trailing dot that copies to your own directory).

cp /projects/ec34/in-biosx000/SNP_calling/BAM_files/ .*
and
cp /projects/ec34/in-biosx000/SNP_calling/reference/ .*

After you are done, it should look roughly like this

```
[ec-bastiaas@login-4  walrus_SNP_calling]$ ls -l
total 11323
-rwxrwx--- 1 ec-bastiaas ec34-member-group    185 Oct 10 10:42 Orosv1mt.dict
-rwxrwx--- 1 ec-bastiaas ec34-member-group  16637 Oct 10 10:42 Orosv1mt.fasta
-rwxrwx--- 1 ec-bastiaas ec34-member-group     10 Oct 10 10:42 Orosv1mt.fasta.amb
-rwxrwx--- 1 ec-bastiaas ec34-member-group     93 Oct 10 10:42 Orosv1mt.fasta.ann
-rwxrwx--- 1 ec-bastiaas ec34-member-group  16644 Oct 10 10:42 Orosv1mt.fasta.bwt
-rwxrwx--- 1 ec-bastiaas ec34-member-group     33 Oct 10 10:42 Orosv1mt.fasta.fai
-rwxrwx--- 1 ec-bastiaas ec34-member-group   4143 Oct 10 10:42 Orosv1mt.fasta.pac
-rwxrwx--- 1 ec-bastiaas ec34-member-group   8336 Oct 10 10:42 Orosv1mt.fasta.sa
-rwxrwx--- 1 ec-bastiaas ec34-member-group      0 Oct 10 10:42 Orosv1mt.fasta.validated
-rwxrwx--- 1 ec-bastiaas ec34-member-group    104 Oct 10 10:41 WLR001.Wal_mt.realigned.bai
-rwxrwx--- 1 ec-bastiaas ec34-member-group   7918 Oct 10 10:41 WLR001.Wal_mt.realigned.bam
-rwxrwx--- 1 ec-bastiaas ec34-member-group    104 Oct 10 10:41 WLR002.Wal_mt.realigned.bai
-rwxrwx--- 1 ec-bastiaas ec34-member-group  34602 Oct 10 10:41 WLR002.Wal_mt.realigned.bam
-rwxrwx--- 1 ec-bastiaas ec34-member-group    104 Oct 10 10:41 WLR003.Wal_mt.realigned.bai
-rwxrwx--- 1 ec-bastiaas ec34-member-group   3342 Oct 10 10:41 WLR003.Wal_mt.realigned.bam
-rwxrwx--- 1 ec-bastiaas ec34-member-group    104 Oct 10 10:41 WLR004.Wal_mt.realigned.bai
-rwxrwx--- 1 ec-bastiaas ec34-member-group  11171 Oct 10 10:41 WLR004.Wal_mt.realigned.bam
```

This is a long list, until you see WLRPAC.Wal_mt.realigned.bam

In this directory, you see a number of BAM files with extension *bam: Each of these represents a single individual and for each BAM file we have an index file with a .bai extension (these index files are needed for many computational programs in order to access these files more efficiently). What we want to do is start calling SNPs from these BAM files. So we also need the reference genome: This reference genome is a fasta file called Orosv1mt.fasta, and also comes with several associated index files (.dict, . fasta.amb, .fasta.ann, etc) that start with the same name (Orosv1mt).

Do

*ls -hal *.bam*

Q: You can now see the file size of your different alignment files in a more human readable format. What do you notice and what impact might this have?

We will now use GATK to call SNPs from these files. GATK is a well-known SNP caller, and VERY well supported with a range of manuals. GATK can perform a bewildering number of different tasks. We will only touch upon a few features today, but you can find all the information you need here:
https://gatk.broadinstitute.org/hc/en-us
Also of great value is the “best practices information” with general guidelines on how to do SNP calling correctly. Please note that GATK is designed for humans. If you work on different organisms, you may encounter complexity that is not expected or covered by GATK approaches.

NOTE: while the commands below are simply given one-by-one (simple shell-scripting) you are certainly encouraged to create a single work-flow (for instance using Snakemake). I leave this up to you, but would enjoy seeing such solutions!

We first need to load the program GATK with the following command:

module load GATK/4.3.0.0-GCCcore-11.3.0-Java-11

We follow the three stage SNP calling as implemented in GATK:
First, you create an individual "pre-called" g.vcf.gz file that contains SNPs per individual.
Second, you place these files into a database.
Third, you call SNPs *simultaneously* for all individuals at once. The reason for this is that GATK uses a statistical model that allows population data to help with assessing the quality of a SNP in each individual. For instance, if a certain allele is numerically prevalent in the entire "population" of samples, GATK takes such a higher probability into account while estimating the statical likelihood of detecting that variant in specific specimens.

We start with the **first step:**
Make use you are in your own walrus directory:

Make a NEW directory called gvcf.

mkdir gvcf

Now run the following command (simply copy paste these three strings together and press <enter>) Make sure you have all of these lines (NOTE: if you type this, the -I in the command is a capital "I" (for input) not a lower case "L"):

*for f in $(ls *.bam); do gatk HaplotypeCaller -R Orosv1mt.fasta *
*-I $f --ploidy 1 -O gvcf/$f.gvcf.gz -ERC GVCF 2> gvcf/HaploC_$f.out; *
echo "$f HaplotypeCalled";done

This should run for 5 to 10 minutes. HaplotypeCaller calls germline SNPs and indels via local re-assembly of haplotypes. While this is running try to understand the ins and outs of the command. What you do is a running a BASH loop so that each BAM file gets analyzed sequentially:

for f in $(ls *.bam); *this part assigns the file name to a variable called $f*
do gatk HaplotypeCaller; *here you tell GATK to run HaplotypeCaller*
-R Orosv1mt.fasta; *this tells GATK which reference to use*
-I $f; *this tells GATK to run a specific BAM file*
--ploidy 1; *this is specific for mitogenomes; can you **understand why?***
-O gvcf/$f.gvcf.gz; *here you create an output file for EACH bam file in the **gvcf** directory*
2> gvcf/HaploC_$f.out; *redirects program output into an out file in the **gvcf** directory*
-ERC GVCF; *tells the program to create GVCF compatible files.*
echo "$f HaplotypeCalled"; *echo to show that you have analysed a BAM file on screen.*
done; *closing the bash **for loop.***

Q: Why do we use ploidy 1 in this script?

Once you see:
'WLRPAC.Wal_mt.realigned.bam HaplotypeCalled' GATK HaplotypeCaller is finished.

(NOTE: If this running takes too long (when all are running simultaneously) you can copy these files into your gvcf directory from here: */projects/ec34/in-biosx000/SNP_calling/gvcf*)

Move to the gvcf directory and list the contents:

cd gvcf
ls

You should see *.out *.gvcf.gz, and *.gvcf.gz.tbi files.
*.out; *contains the program output. You can use cat <filename> to have a look at one of these files and see what GATK has been doing, what command has been run, how long it took, how much memory was required and several other details on what has been done with each file.*
*.gvcf.gz; *compressed text file that contains the initial genotype calls.*
*.gvcf.gz.tbi; *index file for the *.gvcf.gz file, cannot be opened directly.*

We now proceed to the **second step.** Make sure you are still in the gvcf directory. In order to assign all the .gvcf.gz files to GATK, we first make a list of the files we want to analyze:

*ls *.gz > gvcf.list ;* This creates a text file with all the HaplotypeCalled.gvcf.gz filenames.

Then run (copy/paste both strings again):

gatk GenomicsDBImport -V gvcf.list --genomicsdb-workspace-path Walrus_DB \
--intervals NC_004029.2

This command imports the single-sample GVCFs into GenomicsDB before joint genotyping. The import takes about 2-5 seconds. You should now see a Walrus_DB directory appear.

Q: What does NC_004029.2 refer to?

Now we proceed to **step 3**

gatk GenotypeGVCFs -R ../Orosv1mt.fasta -V gendb://Walrus_DB -O Walrus_MT.vcf.gz

The program gatk GenotypeGVCFs performs joint genotyping on one or more samples pre-called with HaplotypeCaller. For these data this also runs in a few seconds, which is WHY we

used a mitogenome. ***If you run this on a nuclear genome with lots of individuals, calling can easily take 24 hours PER chromosome.***

You now have a raw VCF file which contains called genotypes from all individuals you analyzed. Let's have a quick look at this file. Use

zless -S Walrus_MT.vcf.gz **(NOTE type <Q> to exit "less")**

Lines that start with *##* are the header: this contain information on the file format and several other technical details that show what GATK has been calculating and what has been performed.

```
##fileformat=VCFv4.2
##ALT=<ID=NON_REF,Description="Represents any possible alternative allele not already represented at this location by REF and ALT">
##FILTER=<ID=LowQual,Description="Low quality">
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=MIN_DP,Number=1,Type=Integer,Description="Minimum DP observed within the GVCF block">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification">
##FORMAT=<ID=RGQ,Number=1,Type=Integer,Description="Unconditional reference genotype confidence, encoded as a phred quality -10*log10 p(genotype call is wrong)">
##FORMAT=<ID=SB,Number=4,Type=Integer,Description="Per-sample component statistics which comprise the Fisher's Exact Test to detect strand bias.">
##GATKCommandLine=<ID=GenomicsDBImport,CommandLine="GenomicsDBImport --genomicsdb-workspace-path Walrus_DB --variant gvcf.list --intervals NC_004029.2 --genomicsdb-segment-size 1048576 --genomicsdb>
##GATKCommandLine=<ID=GenotypeGVCFs,CommandLine="GenotypeGVCFs --output Walrus_MT.vcf.gz --variant gendb://Walrus_DB --reference ../Orosv1mt.fasta --include-non-variant-sites false --merge-input-in>
##GATKCommandLine=<ID=HaplotypeCaller,CommandLine="HaplotypeCaller --sample-ploidy 1 --emit-ref-confidence GVCF --output gvcf/WLR001.Wal_mt.realigned.bam.gvcf.gz --input WLR001.Wal_mt.realigned.bam>
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=BaseQRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth; some reads may have been filtered">
##INFO=<ID=END,Number=1,Type=Integer,Description="Stop position of the interval">
##INFO=<ID=ExcessHet,Number=1,Type=Float,Description="Phred-scaled p-value for exact test of excess heterozygosity">
##INFO=<ID=FS,Number=1,Type=Float,Description="Phred-scaled p-value using Fisher's exact test to detect strand bias">
##INFO=<ID=InbreedingCoeff,Number=1,Type=Float,Description="Inbreeding coefficient as estimated from the genotype likelihoods per-sample when compared against the Hardy-Weinberg expectation">
##INFO=<ID=MLEAC,Number=A,Type=Integer,Description="Maximum likelihood expectation (MLE) for the allele counts (not necessarily the same as the AC), for each ALT allele, in the same order as listed>
##INFO=<ID=MLEAF,Number=A,Type=Float,Description="Maximum likelihood expectation (MLE) for the allele frequency (not necessarily the same as the AF), for each ALT allele, in the same order as liste>
##INFO=<ID=MQ,Number=1,Type=Float,Description="RMS Mapping Quality">
##INFO=<ID=MQRankSum,Number=1,Type=Float,Description="Z-score From Wilcoxon rank sum test of Alt vs. Ref read mapping qualities">
##INFO=<ID=QD,Number=1,Type=Float,Description="Variant Confidence/Quality by Depth">
##INFO=<ID=RAW_MQandDP,Number=2,Type=Integer,Description="Raw data (sum of squared MQ and total depth) for improved RMS Mapping Quality calculation. Incompatible with deprecated RAW_MQ formulation.>
##INFO=<ID=ReadPosRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt vs. Ref read position bias">
##INFO=<ID=SOR,Number=1,Type=Float,Description="Symmetric Odds Ratio of 2x2 contingency table to detect strand bias">
##contig=<ID=NC_004029.2,length=16565>
##source=GenomicsDBImport
##source=GenotypeGVCFs
```

Have a look at the following line:
##contig=<ID=NC_004029.2, length=16565>

This describes the size of the mitochondrial genome for Atlantic walrus. Google “mammal mitogenome size”.
Q: Would you say that the walrus mitogenome is of expected size for a mammalian mitogenome? Use Google to get your answer

The line starting with:
#CHROM
Contains the column headers of the data in lines below. Each individual variant is found on a chromosome (NC_004029.2), at a certain position (POS), with a reference nucleotide (REF) and variant nucleotide (ALT) with a certain quality score (QUAL) and so forth.

Q: What is the position of **the second variant** to be found, what are the reference and alternative nucleotides, and what is its quality? If you look at the range of qualities scores, with higher values having higher quality, does this variant have a high quality?

Q: What is the position of **the last variant** to be found, what are the reference and alternative nucleotides, and what is its quality? If you look at the range of qualities scores, with higher values having higher quality, does this variant have a high quality?

Then you see in columns what each individual has in terms of GENOTYPE. This is somewhat difficult to see as the INFO and FORMAT columns contain a lot of info.
Each individual genotype records a number of fields separated by “:”

GT:AD:DP:GQ:PL

GT: The actual genotype for that individual. (0:0 means homozygote reference)
AD: is allele depth, the reads supporting each allele.
DP: is depth, number of quality controlled reads.
GQ: genotype quality, higher is better
PL: Phreds scaled likelihood of the genotype being correct, lower is better. NOTE this info is ALSO in the header!

(NOTE type <Q> to exit “less”)

The GATK website has detailed information on how these VCF files are formatted and what they mean.

Now we will run some preliminary data analyses on your raw VCF. We will iteratively change some parameters and see what the impact is on what we call a **Principal Component Analysis (PCA).** PCA is which is a wide-spread statistical method commonly used in population genetics to identify structure in the distribution of genetic variation. PCAs are simple to interpret;

individuals that cluster together are genetically more related than those that cluster away from each other. This is a great way to initially explore your data, to look for statistical outliers and check your data for consistency. Through such iterative approach you can get a feeling for how different filtering settings are impacting your results, and how robust your results are. If results are robust in the face of a range of different filtering settings, your results are likely biologically relevant. If not, you should obviously be really careful that your filtering has been correct.

In order to run this analysis, we'll use a script that is build around a program called smartPCA: (https://github.com/chrchang/eigensoft/blob/master/POPGEN/README)

In order to access the script you'll first need to add a directory to your path:

export PATH="/projects/ec34/in-biosx000/SNP_calling/scripts:$PATH"

Now run in your gvcf directory:

Run_PCA Walrus_MT.vcf.gz

You'll have some warnings flashing by:
Warning: Expected at least 2 parts in INFO entry: ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
Warning: Expected at least 2 parts in INFO entry: ID=MLEAC,Number=A,Type=Integer,Description="Maximum likelihood expectation (MLE) for the allele counts (not necessarily the same as the AC), for each ALT allele, in the same order as listed">
These can be ignored!

The ending should look like this:
338 variants loaded from .bim file.
45 people (0 males, 0 females, 45 ambiguous) loaded from .fam.
Ambiguous sex IDs written to PCA_Walrus_MT.vcf.gz.pruned.nosex .
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 45 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.999869.
338 variants and 45 people pass filters and QC.
Note: No phenotypes present.
--make-bed to PCA_Walrus_MT.vcf.gz.pruned.bed + PCA_Walrus_MT.vcf.gz.pruned.bim + PCA_Walrus_MT.vcf.gz.pruned.fam ... done.
PCA run on Walrus_MT.vcf.gz
Check results in PCA_Walrus_MT.vcf.gz.tab

This output give you information on how many variants are in the file, how many individuals (people, plink is written for humans) and how many SNPs were genotyped and so forth.

Open the PCA_Walrus_MT.vcf.gz.tab:

cat PCA_Walrus_MT.vcf.gz.tab

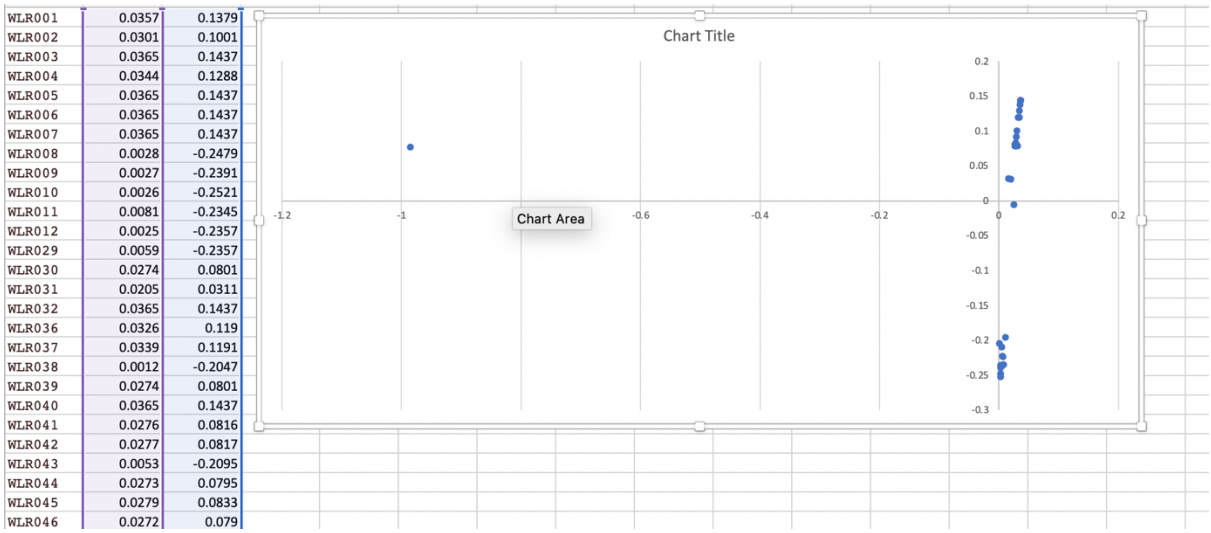
You should see something this:

WLR001	0.0357	0.1377
WLR002	0.0301	0.1000
WLR003	0.0365	0.1436
WLR004	0.0344	0.1287
WLR005	0.0365	0.1436
WLR006	0.0365	0.1436
…		
...		
WLR079	0.0278	0.0822
WLR080	0.0272	0.0789
WLRPAC	-0.9857	0.0768

Now copy paste the lines from your screen starting with WLR001 to WLRPAC to excel to visualize the PCA in a two-dimensional graph. A PCA will cluster individuals that are genetically more similar closer together. Different clusters or groups usually represent biologically relevant populations that are independent, and found in different geographical areas. A PCA usually will separate these individuals along two different axes, which is a standard way to summary thousands of points. In this case, we only have 45.

DISCLAIMER: Normally you would NOT want to use excel at all, but in this case this is a short-cut to quickly see how the visual clustering of the PCA is linked to the data for each specimen and I know you all have access excel or similar. You MAY need use paste special in order for excel to accept the tab divided correctly. Moreover, depending on your regional settings, the columns MAY not be recognized as numerical values. Then use find/replace . for , in order to make excel recognize these as numerical values. You are also welcome to copy the PCA_Walrus_MT.vcf.gz.tab file to your computer and use R to plot this!

In excel (or R), select column 2 and 3 and make a scatterplot. You should get something like below, in which each dot is representing a single individual.



WLR001	0.0357	0.1379
WLR002	0.0301	0.1001
WLR003	0.0365	0.1437
WLR004	0.0344	0.1288
WLR005	0.0365	0.1437
WLR006	0.0365	0.1437
WLR007	0.0365	0.1437
WLR008	0.0028	-0.2479
WLR009	0.0027	-0.2391
WLR010	0.0026	-0.2521
WLR011	0.0081	-0.2345
WLR012	0.0025	-0.2357
WLR029	0.0059	-0.2357
WLR030	0.0274	0.0801
WLR031	0.0205	0.0311
WLR032	0.0365	0.1437
WLR036	0.0326	0.119
WLR037	0.0339	0.1191
WLR038	0.0012	-0.2047
WLR039	0.0274	0.0801
WLR040	0.0365	0.1437
WLR041	0.0276	0.0816
WLR042	0.0277	0.0817
WLR043	0.0053	-0.2095
WLR044	0.0273	0.0795
WLR045	0.0279	0.0833
WLR046	0.0272	0.079

Q: Explore and interpret your graph, also by flicking between the values and the visualization. What do you see?

Obviously, we have not yet filtered these SNP calls at all. As **data coverage** is the most important determinant in the quality of genotype call (you did follow the lecture☺), let us check the read coverage of the individuals we analyze. For this, we use a program all VCFtools. We load the module for VCFtools:

module load VCFtools/0.1.16-GCC-11.3.0

Then we type (or copy paste)

vcftools --gzvcf Walrus_MT.vcf.gz --depth --out Walrus_coverage

This command calculates the average coverage for each of the variants in the datafile, per individual and places these in the **file Walrus_coverage.idepth**. This is a simple text file you can immediately open; open the file Walrus_coverage.idepth:

cat Walrus_coverage.idepth

What you obtain on the screen is the average fold-coverage (MEAN_DEPTH), or the number of reads that -*on average*- are supporting the different SNP (and indel) variants. N_SITES indicates the number of variants and INDV each individual.

INDV	N_SITES	MEAN_DEPTH
WLR001	344	0.517442
WLR002	344	2.3314
WLR003	344	0.0552326
WLR004	344	0.781977
WLR005	344	0.0581395
WLR006	344	0.00290698
WLR007	344	0.0755814
WLR008	344	19.0291
…		
…		
Etc.		

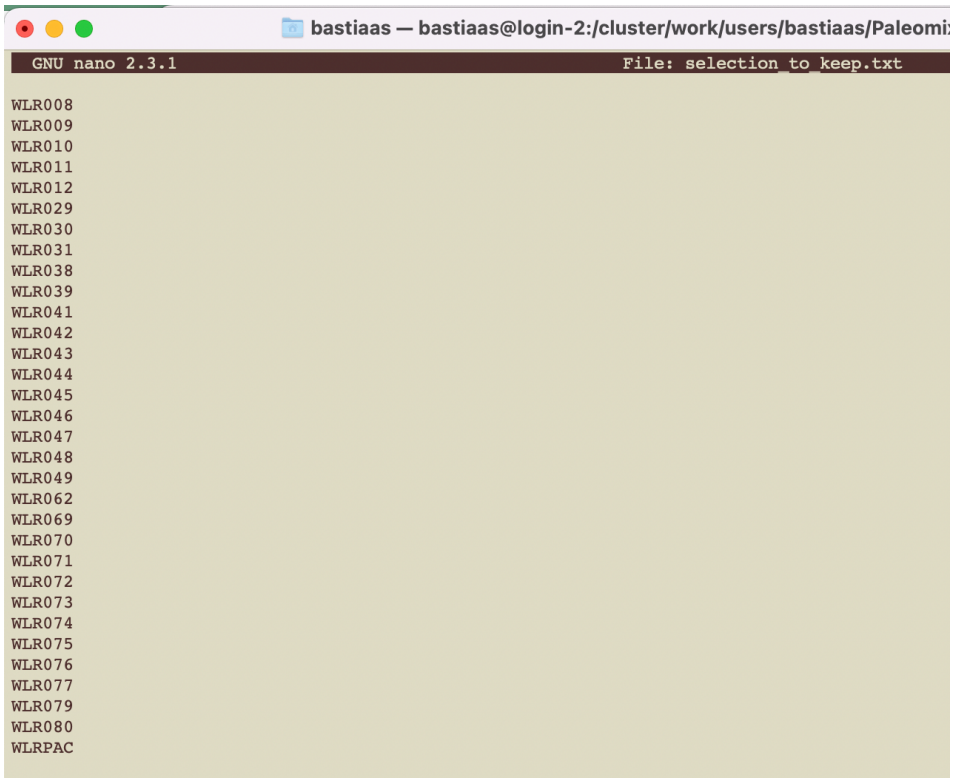
Q: Have a look at this file. What do you see in term of read depth for the different individuals?

We'll copy paste this next the PCA data in excel and compare the output of the PCA with the coverage. What do you see?

Now we will select only those individuals with more **than 3-fold coverage.** What we want is a list of individuals that only contains those specimens with more than 3-fold coverage. You can do this in Excel using different approaches (or manual selection). I leave this up to you how to do that, but let us know if you struggle. Once you have the list, create a file on the Fox cluster with a list of those individuals:

nano selection_to_keep.txt

Paste your individuals into the window.



Exit by hitting <ctrl> <c> simultaneously and <enter>

We will now create a VCF file that contains only those individuals with higher coverage:

*vcftools --gzvcf Walrus_MT.vcf.gz --keep selection_to_keep.txt *
--recode --recode-INFO-all --stdout | gzip -c > Walrus_MT_3X.vcf.gz

This command creates a new VCF file called Walrus_MT_3X.vcf.gz.

Run the PCA analyses again on this new VCF file:

Run_PCA Walrus_MT_3X.vcf.gz

And

cat PCA_Walrus_MT_3X.vcf.gz.tab

Now copy paste the lines in a new tab in excel, from your screen starting with WLR008 to WLRPAC to visualize the PCA in a two-dimensional graph again

Q: Compare the PCA that you get with the previous PCA. What is the same and what is different?

Q: Based on your quick analysis, do you think the clusters are depending on sequencing coverage?

Now another way of making sure we have SNP and genetic variants in our data, is to only select those SNPs that we observe **in multiple individuals**. At the moment, we may have SNPs in our data that we have only observed once, in a single individual. Such *rare* variants could be driven by sequencing errors, batch processing errors or something else. It is possible to filter out such variants by restricting our analyses to be looking at common variants (i.e. genetic SNPs that are observed in multiple individuals. For this, we can filter using something called **a Minor Allele Frequency (MAF) threshold**. A Minor allele frequency (MAF) is the frequency at which the second most common allele occurs in a given population. If you for instance put a MAF at 0.1, you want to observe the less common allele at least 10% of the time. In our dataset here (with now 32 specimens in the dataset) this means we would want to see such variation in at least 3.2 specimens (or in this case then 4 specimens) as you cannot have 0.2 of a specimen.

Again, we will use the program VCFtools to filter such variants.

Run:
vcftools --gzvcf Walrus_MT.vcf.gz --keep selection_to_keep.txt \
--maf 0.1 --recode --recode-INFO-all --stdout | gzip -c > Walrus_MT_MAF.vcf.gz

Now Run the PCA analyses again on this new VCF file:

Run_PCA Walrus_MT_MAF.vcf.gz

And

cat PCA_Walrus_MT_MAF.vcf.gz.tab

Now copy paste the lines in a new tab in excel again, from your screen starting with WLR008 to WLRPAC to visualize the PCA in a two-dimensional graph.

Q: Compare the PCA that you get with the previous PCA's. What is the same and what is different? Can you explain what has changed and why, given your filtering?

Lets go back to the second PCA, the one where we did not include a MAF filter and think about the reference specimens:
WLR008 (reference Svalbard)
WLR009 (reference Svalbard)
WLR010 (reference Svalbard)
WLR011 (reference Svalbard)
WLR012 (reference Svalbard)
WLR030 (reference Greenland)
WLR031 (reference Greenland)

Q: Where do the control specimens cluster?

Q: How would you interpret then the whereabouts of the other historical walrus specimens? Is this dependent on any of the filtering we did? What does this mean for our ideas on walrus trade from Norse Greenland?

You have reached the end of the exercise! The scenario you were given is hypothetical, and some details on the distribution of specimens and data were simplified. Nonetheless, the data you have worked with are in fact true data that were used to investigate the walrus trade in quite some detail. If you are interested you can read how we used these data to paint quite a detailed picture of this historic trade, which crossed continents and putatively ended up all the way from Greenland to Kyiv, and from there on to Eurasia.

2018
Ancient DNA reveals the chronology of walrus ivory trade from Norse Greenland
https://royalsocietypublishing.org/doi/full/10.1098/rspb.2018.0978

2020
Ecological globalisation, serial depletion and the medieval trade of walrus rostra
https://www.sciencedirect.com/science/article/pii/S0277379119305736

2022
Walruses on the Dnieper: new evidence for the intercontinental trade of Greenlandic ivory in the Middle Ages
https://royalsocietypublishing.org/doi/full/10.1098/rspb.2021.2773

2024
Greenland Norse walrus exploitation deep into the Arctic
https://www.science.org/doi/10.1126/sciadv.adq4127