Population Structure of Eurasian Otters in Fennoscandia

Through Analysis of Single Nucleotide Polymorphisms

Luis Perlich

Degree project • 15 credits
Swedish University of Agricultural Sciences, SLU
Faculty of Forest Sciences / Department of Fish, Wildlife and Environmental Studies
Forest and Landscape / Biology
Examensarbete / SLU, Institutionen för vilt, fisk och miljö
2025:5
Umeå 2025

Population Structure of Eurasian Otters in Fennoscandia. Through Analysis of Single Nucleotide Polymorphisms

Populationsstruktur inom den Fennoskandinaviska Utterstammen. Genom Analys av Single Nucleotide Polymorphisms

Luis Perlich

Supervisor:	**Anita Norman, Swedish University of Agricultural Sciences, Department of Fish, Wildlife and Environmental Studies**
Examiner:	Therese Löfroth, Swedish University of Agricultural Sciences, Department of Fish, Wildlife and Environmental Studies

Credits:	15 credits
Level:	First cycle, G2E
Course title:	Independent Project in Biology
Course code:	EX0997
Programme/education:	Forest and Landscape
Course coordinating dept:	Department of Fish, Wildlife and Environmental Studies
Place of publication:	Umeå
Year of publication:	2025
Series:	Examensarbete / SLU, Institutionen för vilt, fisk och miljö
Serial Number:	2025:5
Cover picture:	Eurasian otter in Norway

Keywords:	Lutra, Eurasian otter, SNP, population structure, genetics, IBD, Finland, Norway, Sweden

Swedish University of Agricultural Sciences
Faculty of Forest Sciences
Department of Fish, Wildlife and Environmental Studies
Molecular Ecology Group

Abstract

Eurasian otters are an essential species within Palearctic freshwater ecosystems. Monitoring populations through non-invasive genetics can provide helpful information advising wildlife management. An investigation of population structure within the Fennoscandian metapopulation of otters was conducted through the analysis of 63 SNP-loci each from 93 individual DNA-samples collected from Eurasian otters in Finland, Norway, and Sweden. Otters in Western Norway exhibited distinct genetic differentiation in relation to the rest of the sample group. Finnish otters showed higher genetic variation, while Swedish otters held an intermediary genetic profile between Finnish and Norwegian populations. Furthermore, otters in Eastern Norway were genetically more similar to the Swedish population than to other Norwegian otters. Possible genetic traces of reintroductions of Norwegian otters into struggling Swedish populations in the 20th century were detected. Isolation by Distance (IBD) was identified in the metapopulation.

Keywords: Lutra, Eurasian otter, SNP, population structure, genetics, IBD, Finland, Norway, Sweden

Table of contents

List of tables

List of figures

Abbreviations

Abbreviation	Description
DAPC	Discriminant Analysis of Principal Components
HWE	Hardy-Weinberg Equilibrium
IBD	Isolation By Distance
IUCN	International Union for the Conservation of Nature
MDG	Majority Descent Group
PCA	Principal Component Analysis
SNP	Single Nucleotide Polymorphism
sPCA	Spatial Principal Component Analysis

1. Introduction

The Eurasian otter (*Lutra lutra*, L., 1758) is mustelid species native to the Palearctic (Loy et al. 2022). The species can be classified as the mammalian apex predator within freshwater ecosystems in Europe, but its habitat also includes coastal wetlands, lagoons, and marine shores (Kruuk 2006). At present it is listed as Near Threatened (NT) on the IUCN Red List (Loy et al. 2022) the EU Habitats Directive (Directive 1992), and is listed in Appendix II of the Bern Convention on the Conservation of European Wildlife and Natural Habitats (Council of Europe 1979). These classifications are a result of population decline during the 20th century caused by environmental pollution and strong persecution as a furbearing species. However, populations are recovering, through strict protections, reduction of environmental pollutants, and reintroduction programs (Loy et al. 2022). Nowadays, the Eurasian otter is again present in nearly all of Fennoscandia (Loy et al. 2022), yet its elusive nature and its aquatic lifestyle complicate the detection of its presence. To aid recovery efforts in Southern and Central Sweden, both captive-bred Swedish as well as wild-caught Norwegian otters were released in Södermanland and Uppland (Sjöåsen 1996), leading to potential blending of genetics.

Using genetics to gather information on a species is a functional tool in conservation and research. It is an ethical and non-invasive method where DNA is extracted from substrates such as hair, scat, or tissue and blood from deceased individuals. It is also easier to attain larger sample sizes due to the ease of collecting large amounts of those substrates, improving reliability of the results. Using these types of data, we are able assess population health and structure through genetic variation, genetic exchange, genetic drift, and genetic relatedness (Allendorf et al. 2012). All these metrics are supplementary knowledge improving decisions in wildlife management, especially concerning extinction risk or population fitness.

Genomic markers such as Single Nucleotide Polymorphisms (SNPs), denoting one base change with two possible nucleotides in a DNA sequence, that can occur in both coding or non-coding regions, are a useful tool to assess a populations genetic structure. They allow for better comparison of data between laboratories, as they are less prone to genotyping errors in comparison to microsatellites (Vignal et al. 2002). In the genome, SNPs are numerically the most common type of polymorphism (Allendorf 2012), allowing a high genomic resolution.

The analysis of genetic population structure can provide crucial information for the conservation and management of a given species by helping to identify potential management units (Manel et al. 2003). The genetic variation or distance within a species is affected by their dispersal capacity, as gene flow reduces genetic differences through admixture, therefore genetic distance is often visible

within isolated subpopulations (Allendorf et al. 2012). It is generally accepted that genetic variance and diversity is beneficial for a species' resilience towards environmental changes (Allendorf et al. 2012) (source). However, a certain degree of inbreeding or a less varied genetic profile does not necessarily equate bad fitness. Instead, it can be a sign of local or niche adaptation that can be lost through outbreeding depression (Ralls et al. 2001).

Eurasian otters' social organization seems to align with most other mustelid species, in that individuals have loose territories/home ranges that they defend only against the same sex and overlaps of territories of the opposite sex (Kruuk 2006). Females exhibit philopatric behaviour while males disperse over larger distances (Quaglietta et al. 2013). In addition, a significant subset of vagrant individuals may not be integrated into the reproductive population (Kruuk 2006). While otter dispersal in general is linked to the presence of watercourses, which are inherently non-linearly distributed in the landscape, otters have shown astounding long-distance dispersal capability. However, some populations exhibit isolation by distance (IBD), for example in Great Britain (Dallas et al 2002). IBD describes a statistical approach to test the relationship between geographic distance and genetic differentiation between populations, i.e. whether populations in close geographic proximity are more similar genetically than those further apart. The knowledge that the population in Fennoscandia is made up not just of geographically distinguishable but also genetically distinguishable groups, as well as areas of admixture, are factors that should be considered in their conservation, and has been previously explored in several studies (Honnen et al. 2015, Tison et al. 2015).

The aim of this thesis project is to use SNP-markers to detect population structure within the larger Fennoscandian population and test for the presence of IBD, as SNPs will provide a more comparable basis for future studies. The Fennoscandian geography of a mountain chain dividing the peninsula likely affects the gene flow within the otter population, and possibly causes IBD.

Based on genetic clusters that have already been identified through microsatellite analysis (Honnen et al. 2015, Tison et al. 2015), I hypothesize that higher degrees of gene flow will be found along the Norwegian coastline with some sub clustering due to the distances between north and south, and the mountainous terrain. In addition, higher degrees of gene flow (admixture) are expected around the Bothnian bay and in Lapland. I expect the inland of Scandinavia, i.e. east of the peninsular watershed and Southern Sweden, to show higher degrees of genetic distinction with limited connectivity through major rivers with the otters around the Bothnian bay. In general, the Swedish otters should exhibit a genetic profile that is intermediary between Finnish and Norwegian otters, and Finnish otters should be genetically most varied, not the least because of the Finnish geography of thousands of interconnected lakes.

2. Methods

2.1 Sample Collection, DNA Sequencing and Genotyping

Before the commencement of this thesis work, tissue or blood samples were opportunistically collected from deceased otters (e.g. vehicle collisions) in Finland (n=40), Norway (n=32), and Sweden (n=32). Of these 38, 24, and 32 respectively yielded usable genetic data for a total of 94 individuals included in the study. The sample collection included GPS locations and brief further description of the place of collection, of varying precision.
My supervisor, Anita Norman prepared the genetic data for my analysis.

SNPs were filtered in R-Studio with the following criteria:

1. Single SNP in read.
2. Flanking regions 35 bp (right) and 30 bp (left) in length.
3. Minor allele frequency minimum 0.35.
4. Minimum number of individuals represented in SNP was 46

The same procedure starting from reference genome alignment to SNP filtering was conducted for each of:

1. Whole genome
2. Mitochondrial genome
3. X chromosome
4. Y chromosome

In the end, 300 SNPs were selected for assay development and the original samples were genotyped on the Biomark (Fluidigm, San Francisco, CA, USA). This analysis uses the first plate of 96 SNPs from the larger dataset.

2.2 Data Analysis

2.2.1 Metadata Preparation

Metadata was harmonized for combined analyses. Coordinates belonging to Finnish samples were converted from the ETRS-TM35FIN format to the global standard WGS 84. In the Norwegian sample set all but four individuals were missing GPS coordinates but included the name of the municipality, therefore the precise location was substituted with the coordinates of the centre of the municipality, which were identified using GeoHack (GeoHack n.d.). The Swedish metadata was cleanest, with most coordinates provided in WGS 84 format. One individual, *LuS_14,* had coordinates that would place it east of Stockholm, yet the rest of the location description clearly indicated it was instead located around Fensbol, in Torsby municipality in Värmland County. Therefore, the coordinates were substituted with the location of Fensbol. Overall, the description of the

sampling location was standardized to include only the name of the municipality and country, the smallest common denominators in the sample sets.

2.2.2 Quality Control and Filtering

SNP genotypes were converted to a genind object, and initial quality control was performed using the adegenet package (Jombart and Ahmed 2011) in R-Studio (R 4.4.2, R Core Team 2024). The dataset was examined and filtered for missing data, and filtered to retain only autosomal loci, removing loci associated with mitochondrial DNA or sex chromosomes. Data completeness was assessed using summary statistics and computing missing value distributions across populations and individuals. Heterozygosity was calculated for all individuals and checked for anomalies. The full code is included in Appendix 1.

2.2.3 Genetic Structure Analyses

Hardy-Weinberg Equilibrium

To identify deviations in SNP's from expectations of random-mating and neutrality (not under selection), as well as to provide a baseline model, Hardy-Weinberg Equilibrium (HWE) was assessed for each autosomal locus using exact tests with 999 permutations. Loci significantly deviating from HWE ($p < 0.001$) were identified and visualized using a horizontal bar plot.

Minor Allele Frequency (MAF)

The allele frequencies were calculated by population, and minor allele frequencies (MAF) were derived using the formula MAF = $min\ (p,\ 1 - p)$. The data was checked for potential genotyping issues by visualizing loci by population and sorting them by frequency.

Genetic Distance and Clustering

To assess the gene flow and degree of isolation between the three populations, Nei's genetic distance via dist.genpop (Jombart and Ahmed 2011) was used to compute genetic distances between populations. A hierarchical clustering (hclust, Jombart and Ahmed 2011) dendrogram was constructed and visualized using dendextend (Galili 2015) and ggdendro (DeVries et al. 2024) Furthermore, to help in the presentation and analysis of results the dendrogram was split into sectors, denoted with letters. A table was created to create a better overview of the sectors and clustering.

Principal Component Analysis (PCA)

Autosomal SNPs were scaled using scaleGen (Jombart and Ahmed 2011), and PCA was performed using ade4::dudi.pca (Dray et al. 2023). The first two principal

components were visualized with individual points coloured by population and labelled by individual-ID (e.g. *LuF_01*). PCA is a technique that reduces the complexity of genetic data, transforming it into a smaller set of principal components. Generally, the first few components capture the majority of genetic variation within the population. PCA describes overall genetic variation without using predefined groups. By using PCA on this dataset, one is able to identify clusters in the data that correspond to real-world populations.

Discriminant Analysis of Principal Components (DAPC)

To reinforce the result of the PCA, a DAPC was performed using adegenet::dapc (Jombart and Ahmed 2011) retaining 50 principal components and 2 discriminant functions. Population clustering was visualized using Eigenvalue plots and scatter plots. DAPC reduces data dimensionality by summarizing genetic variation between groups to visualize population structure between these predefined groups.

Spatial Principal Component Analysis (sPCA)

The spatial structure was further explored using adegenet::spca (Jombart and Ahmed 2011), examining the significance of spatial components in the data. The coordinates were jittered to prevent overlapping points, and a Delaunay triangulation was used to define the spatial network. Two global axes were retained, and spatial eigenvalues were plotted. Individual sPCA scores for each axis were mapped geographically, with colour gradients representing spatial genetic patterns.

2.2.4 Spatial and Landscape Genetics

Geographic Mapping

All sampling locations were plotted on a map of Fennoscandia using ggplot2 (Wickham 2016) and sf (Pebesma and Bivand 2023), with relevant geographic features (rivers, lakes, mountain ranges, national boundaries) added using data from Natural Earth (Natural Earth 2021), a public domain geographic database. The data points were coloured by population and jittered to prevent overlap. The labels were added using ggrepel (Slowikowski 2024) and the individual names from the metadata. The created maps visualize distance between samples, some potential geographical obstacles and pathways for otter movements, and help to identify sampling coverage and potential sampling biases. A table was created to visualize geographic cluster assignment.

IBD using Mantel Test

A Mantel test was conducted to assess the correlation between genetic distance (from the PCA) and geographic distance (Haversine), thereby testing for IBD. The test used the Pearson correlation and permutated 999 times. The results were visualized via a scatterplot and a linear regression.

3. Results

3.1 Quality Control and Filtering

The quality control of the data revealed matching numbers for number of included individuals (94) and population sizes, and that only 0.2% of the data was missing. However, one individual (*LuF_01*) accounted for half of the missing data and was therefore removed from the dataset, leading to a total sample size of 93 otters. The filtering for autosomal loci reduced the number of assessed loci from 96 to 80. There were no anomalies across the individual heterozygosity values, indicating sample cross-contamination was unlikely.

3.2 Genetic Structure Analyses

3.2.1 Hardy Weinberg Equilibrium

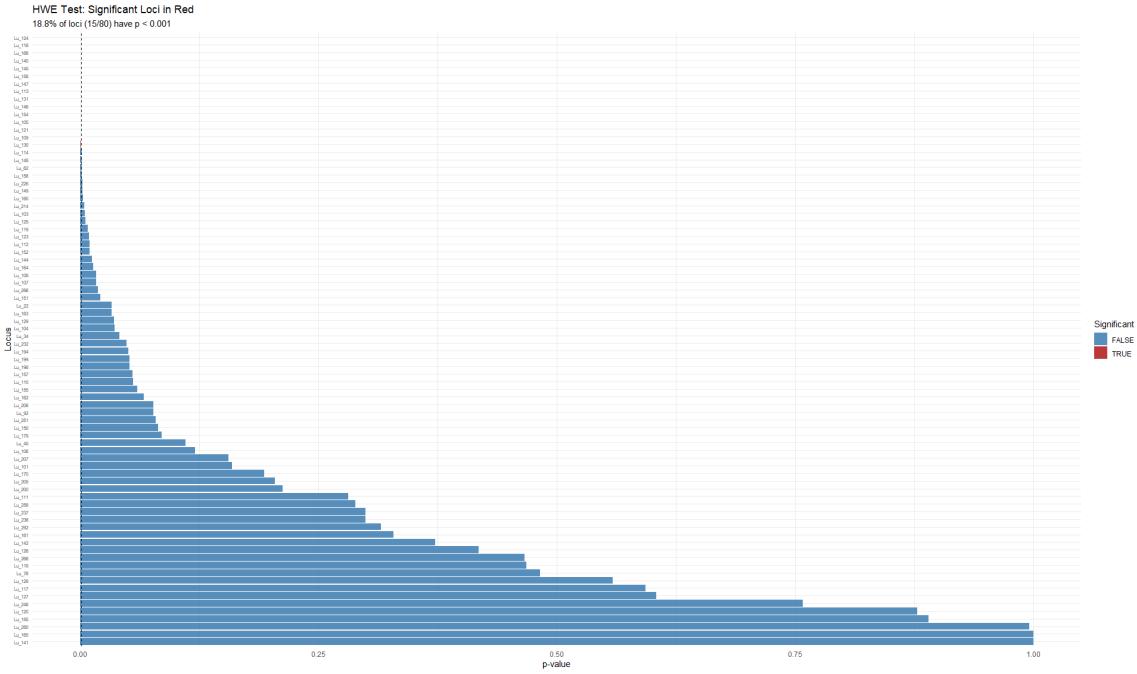

Figure 1 – Loci significantly deviating from HWE (p<0.001). P-values sit on the x-axis, with Loci on the y-axis. Red denotes a significant deviation from HWE and the blue

The test for HWE revealed that 18.8% (15 of 80) of the assessed loci deviated significantly from the proportions expected by HWE-model (Figure 1). An investigation yielded no significant pattern and therefore the 15 loci were removed in order to not cause biases in or affect the downstream analysis. The resulting dataset had 65 loci in total.

3.2.2 Minor Allele Frequency

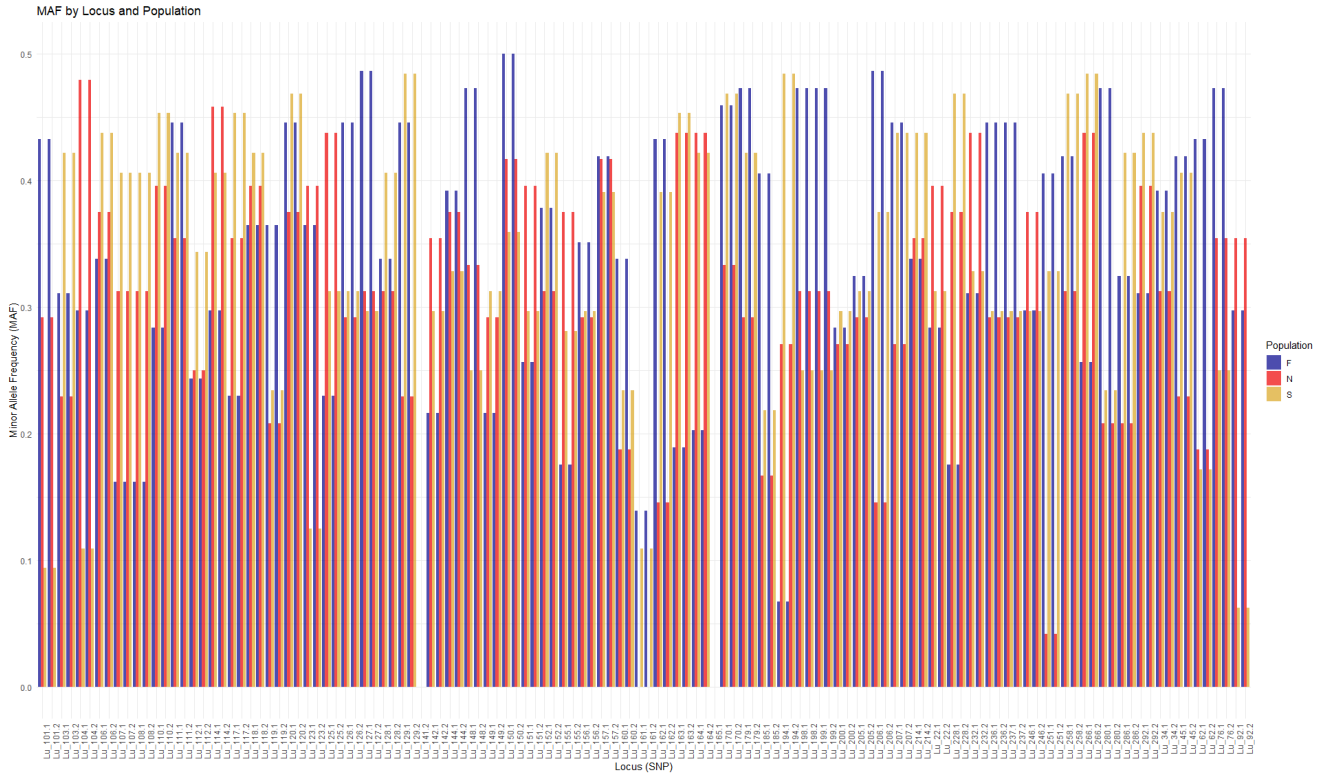

Figure 2 - Minor Allele Frequency (MAF) by locus and population. Loci are on the x-axis and MAF on the y-axis. Blue colour indicates the Finnish population, red Norwegian, yellow Swedish.

The Minor Allele Frequency (MAF) analysis returned valid values for the assessed loci. By plotting the MAF (Figure 2) three loci, namely *Lu_141.2* and *Lu_165.1*, and both alleles of *Lu_161* stick out visually. On closer inspection of the allele frequencies, *Lu_141.2* and *Lu_165.1* returned values of 1.0 indicating these as monomorphic alleles across all populations, while the *Lu_161.1* was monomorphic in the Norwegian population (allele-frequency = 1.0) and *Lu_161.2* not present in the Norwegian population. The loci *Lu_141* and *Lu_165* were excluded from the dataset, resulting in 63 included loci.

3.2.3 Genetic Distance and Clustering

The hierarchical clustering dendrogram (Figure 3) traced the genetic distance between individuals (Table 1). In sectors A and B the dendrogram shows a distinct branch of Norwegian otters, with the individual *LuN_19* in sector A exhibiting differentiation from the rest of the branch. It also shows again the genetic difference of Finnish otters overall in comparison to the Norwegian and Swedish populations, which show a higher degree of closeness with each other than with the Finnish otters. The Finnish branch can be sub-divided into two

major branches (sectors C and D), of which each contains only one otter from the other populations, C one Norwegian (*LuN_29*), and D one Swedish (*LuS_31*).
The branch of Swedish otters is more complex in structure. It can be further divided into Majority Descent Groups (MDG) that are dominated by individuals of a certain population. Sectors E, F, and G sit on one branch with two subgroups. E contains purely Norwegian individuals, F one otter from Norway and one from Sweden, and G is majorly composed of Finnish otters.
According to the MDG's, sectors H, J, K, and L belong to the Swedish descent group, yet sector I is by majority Norwegian. Worthy of note is one outlying Norwegian otter (*LuN_08*) that appears to be closest related to Swedish otters (sector J). Additionally, a Finnish otter(*LuF_21*) outlies in Sector I, which is to 50% composed of Norwegian individuals, and to 40% of Swedish otters. Lastly, sector L also contains individuals from all three populations with a Swedish majority: two Finnish otters, and one Norwegian.

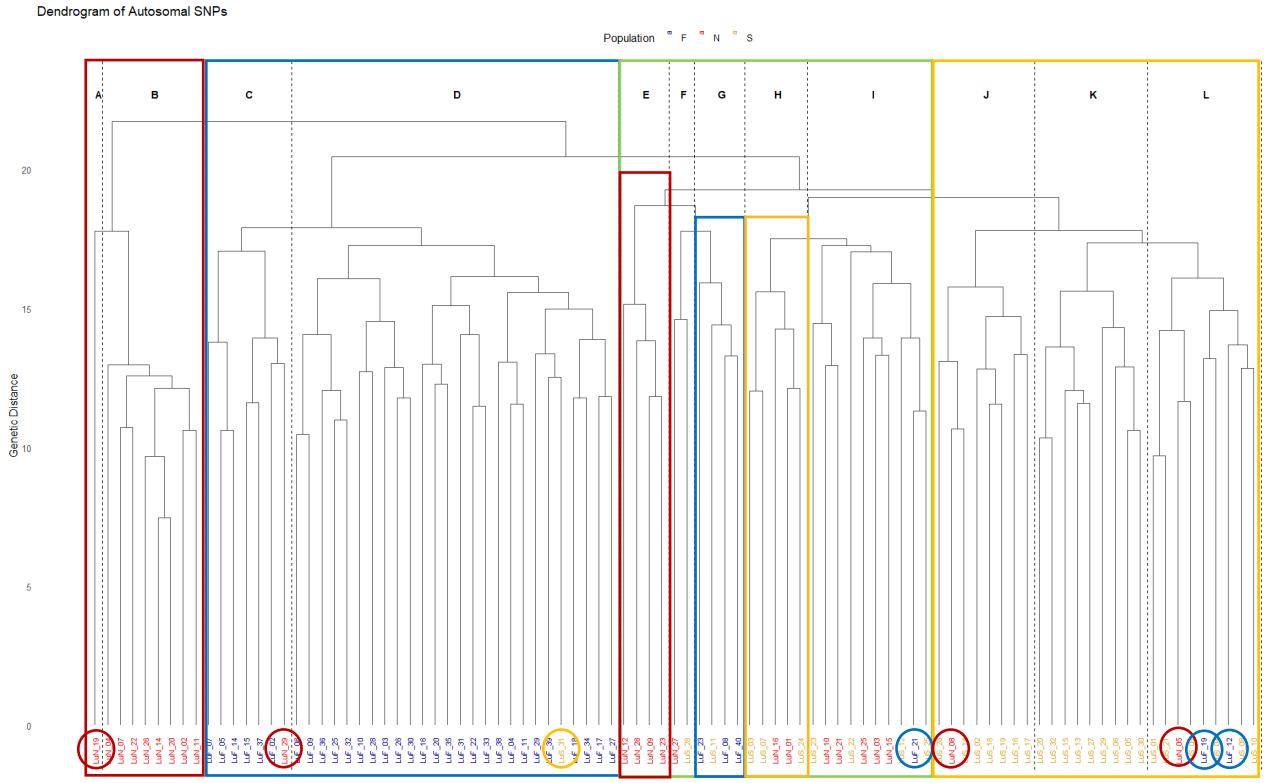

Figure 3 - Hierarchical Clustering Dendrogram of the samples showing the genetic distance between individuals. Denoting population by the colour of the individual label. Blue for Finnish, red for Norwegian, and yellow for Sweden. Through vertical lines sectors are created to aid in the tracing of descent groups, which are further differentiated by coloured rectangles according to populations, and green marking a mixed branch. Outliers are highlighted through circles coloured by population.

Table 1 - Dendrogram sectors in detail, with assignments of branches, Majority Descent Groups, and listing of individuals.

Sector	Branch	Individuals	Majority Descent Group
A	Norwegian	LuN_19	Norwegian
B	Norwegian	LuN_02, 04, 07, 11, 14, 20, 22, 26	Norwegian
C	Finnish	LuF_02, 05, 07, 14, 15, 37, LuN_29	Finnish
D	Finnish	LuF_03, 04, 06, 09, 10, 11, 17, 18, 25, 26, 27, 28, 30, 32, 33, 34, 36, 38, 39 LuS_31	Finnish
E	Mixed	LuN_09, 12, 23, 28	Norwegian
F	Mixed	LuN_27 LuS_28	Norwegian/Swedish
G	Mixed	LuF_08, 23, 40 LuS_11	Finnish
H	Swedish	LuN_01, 16 LuS_03, 07, 24	Swedish
I	Swedish	LuF_21 LuN_03, 10, 15, 21, 25 LuS_12, 22, 25	Norwegian
J	Swedish	LuN_08 LuS_02, 16, 17, 18, 19, 26	Swedish
K	Swedish	LuS_05, 06, 13, 15, 20, 27, 29, 30, 32	Swedish
L	Swedish	LuF_12, 19 LuN_05 LuS_01, 04. 08, 09, 10, 21	Swedish

3.2.4 Principal Component Analysis

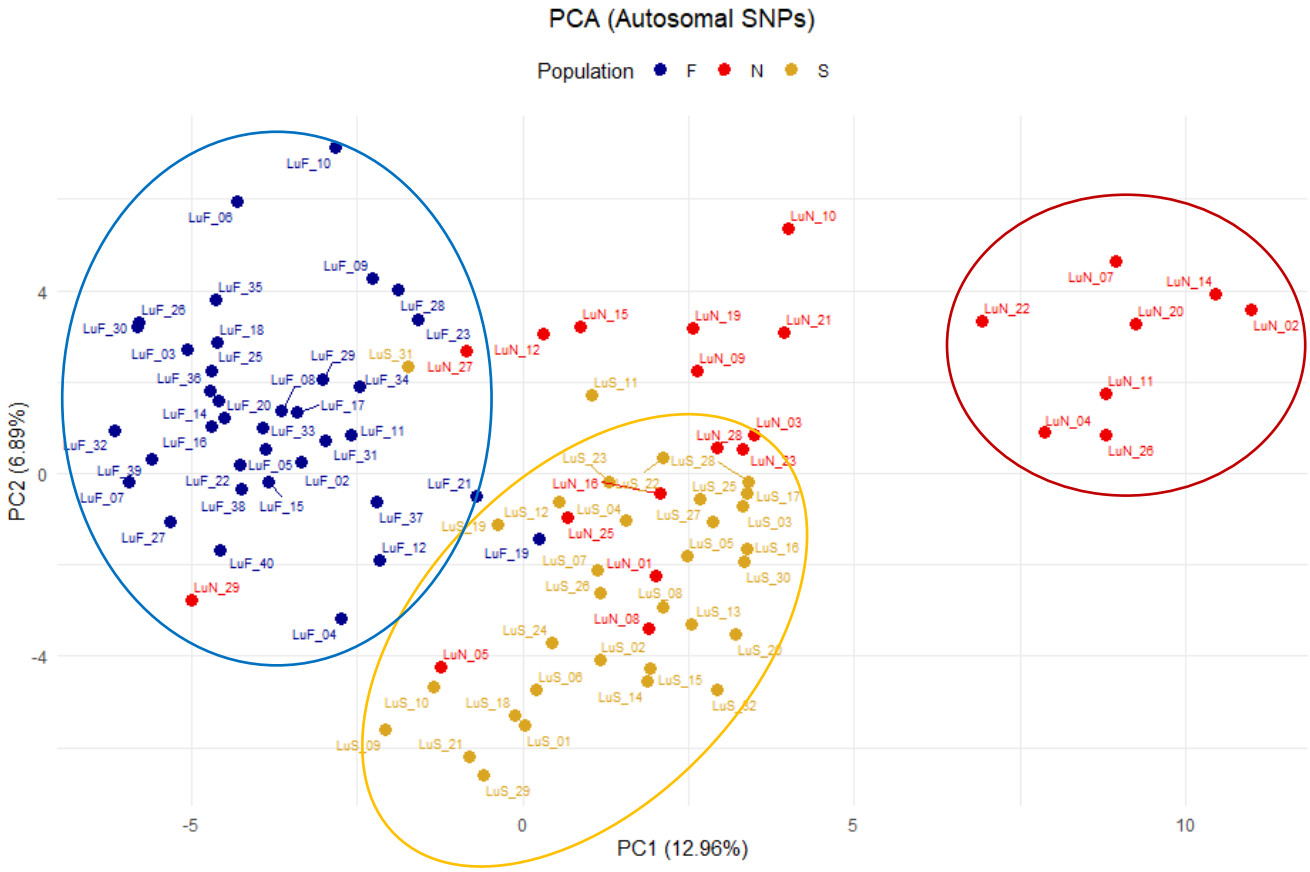

Figure 4 – Principal Component Analysis labelled by individual and coloured by population. The first principal component sits on the x-axis and the second principal component on the y-axis. Ellipses are drawn to highlight clusters.

The PCA mapping shows three diffuse clusters with varying degrees of overlap between the populations (Figure 4). The Finnish population forms a cluster with the samples *LuN_27*, *LuN_29*, and *LuS_31*, and has one outlier *LuF_19* that sits on the edge of the Swedish cluster. The Norwegian population exhibits sub-clustering, with a cluster (*LuN_02, 04, 07, 11, 14, 20, 22, 26*) separate from the rest of the Norwegian otters that are more similar to Finnish and Swedish otters. Especially notable are the individuals *LuN_01, 05, 08, 16,* and *25*, that are more closely related to each other and Swedish otters than to the other Norwegian samples. The Swedish population appears to be intermediary between the Finnish and Norwegian populations, with some otters showing higher difference.

3.2.5 Discriminant Analysis of Principal Components

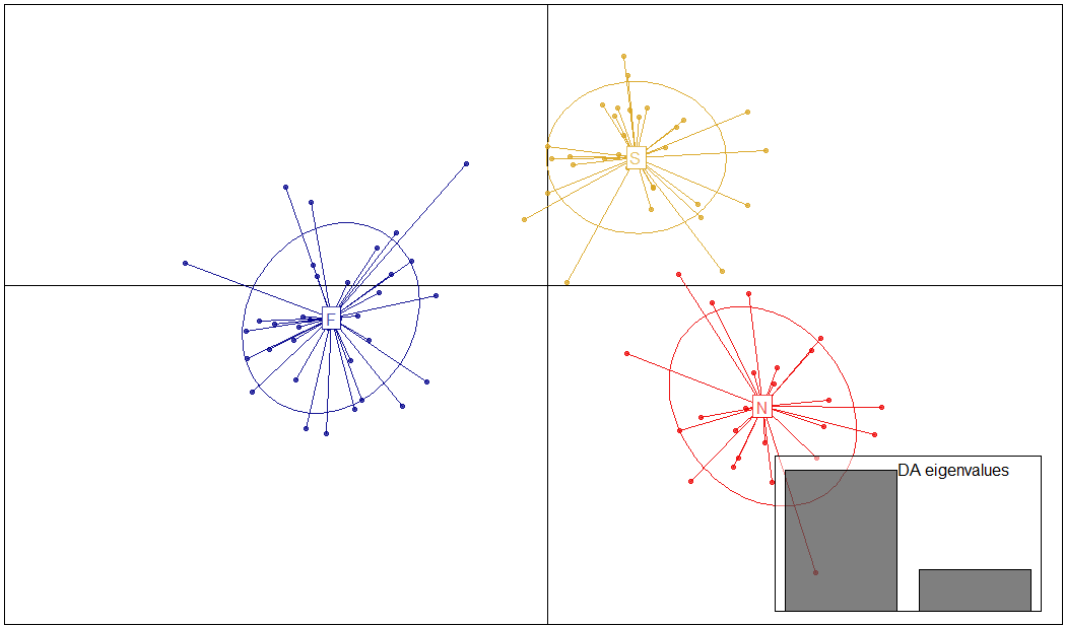

Figure 5 – Scatterplot of DAPC. Points represent individuals; populations are labelled within their 95% inertia ellipses. Coloured by population.

The DAPC shows clear genetic structuring between the three populations (Figure 5), with Norwegian and Swedish otters appearing to be more closely related to each other than to the Finnish population, and the Swedish population being overall closer to the Finnish than the Norwegian is to the Finnish.

3.2.6 Spatial Principal Component Analysis

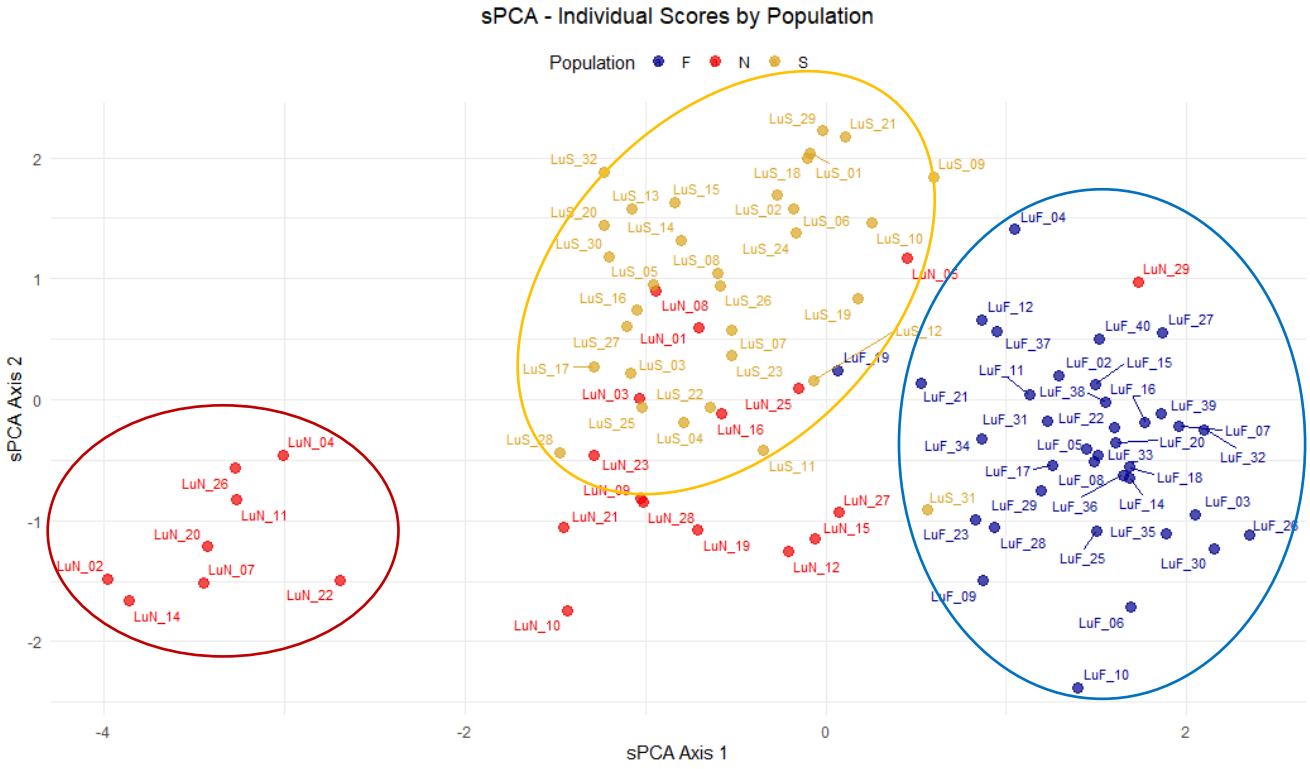

Figure 6 - sPCA Individual scores labelled by individual and coloured by population. The first spatial principal component sits on the x-axis and the second spatial principal component on the y-axis. Ellipses are drawn to highlight clusters.

The sPCA (Figure 6) largely matches the three clusters identified in the PCA, see Figure 4, with mostly the same overlaps and outliers shown through the analysis of the spatial principal components.

3.3 Spatial and Landscape Genetics

3.3.1 Geographic Mapping

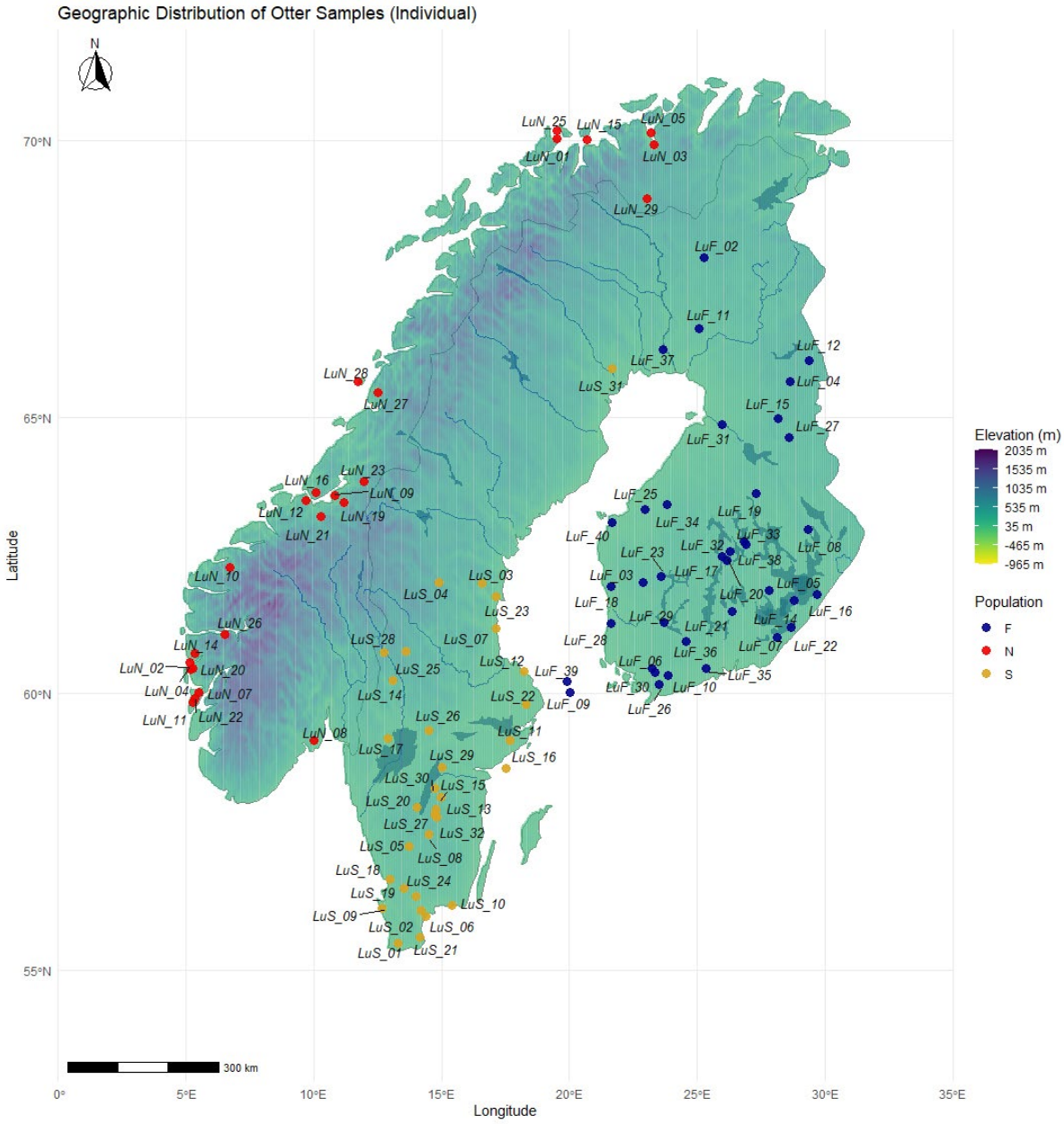

Figure 7 – Geographic Distribution of samples labelled by individual and coloured by population, showing landscape features such as elevation, rivers, and lakes.

Mapping the samples in Fennoscandia revealed several geographical clusters in the sampling (Figure 7). There is a cluster in Western Norway hereby referred to as the Vestland-cluster, one sample in Eastern Norway the Vestføld-outlier (*LuN_08*), a cluster in Central Norway the Trøndelag-cluster, and one in Northern Norway the Finnmark-cluster. In Sweden the sampling is densest in Southern and South-Eastern Sweden and more diffuse further north. For the sake of more

differentiated reference to sub-groups, I am going to refer to every sample south of *LuS_16* and 17 as the Götaland-cluster, north of those as the Svealand-cluster, and everything north of *LuS_12*, *LuS_25*, and *LuS_28* as the Norrland-cluster. In Finland I am going to divide them into Southern and Northern subgroups, dividing below *LuF_27* and *LuF_31*. In Table 2, individual assignment to clusters is specified.

Table 2 - Geographical cluster assignment, with listed individuals.

Key	Individuals
Vestland	LuN_02, 04, 07, 11, 14, 20, 22, 26
Vestføld	LuN_08
Trøndelag	LuN_09, 10, 12, 16, 19, 21, 23, 27, 28
Finnmark	LuN_01, 03, 05, 15, 25
Götaland	LuS_01, 02, 05, 06, 08, 09, 10, 13, 15, 18, 19, 20, 21, 24, 27, 29, 30, 32
Svealand	LuS_11, 12, 14, 16, 17, 22, 25, 26, 28
Norrland	LuS_03, 04, 07, 23, 31
Southern Finland	LuF_03, 05, 06, 07, 08, 09, 10, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 32, 33, 34, 35, 36, 38, 39, 40
Northern Finland	LuF_02, 04, 11, 12, 15, 27, 31, 37

3.3.2 IBD using Mantel Test

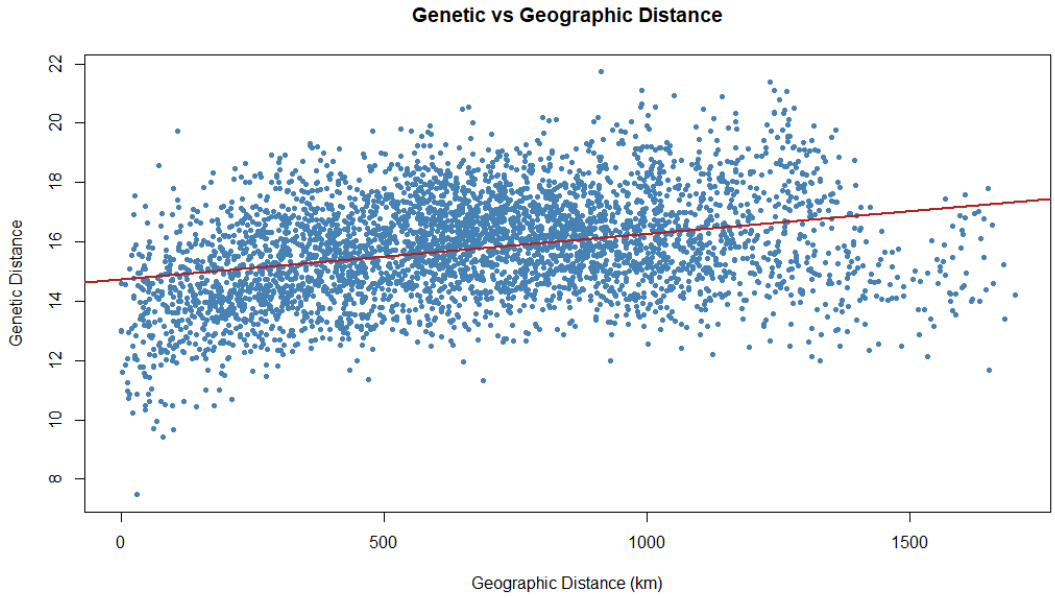

Figure 8 - Scatterplot of geographic vs genetic distance with linear regression to visualize the trend. Geographic distance sits on the x-axis and genetic distance on the y-axis. Data points represent pairwise comparisons between the populations.

The result of the Mantel test returned a p-value of 0.001, and a moderate positive linear correlation between geographic and genetic distance, that becomes visible in the created scatterplot with regression line (Figure 8).

4. Analysis and Discussion

Genetic Structure

The Eurasian otter recolonized Fennoscandia at some point after the last glaciation event, originating from a single small population that survived in a refuge area (Mucci et al. 2010). Out of the three countries included in this study, Finland has the largest interface with other otter populations in Russia and Estonia, therefore Finnish otters could have the most varied genetics within the Fennoscandian population. Norway and Sweden are located on the same peninsula and should exhibit genetic closeness at least east of the Scandinavian mountain chain, when assuming either a colonization from the South (Denmark) or the North (Finland or Russia). Otters disperse along and through watercourses, and although some individuals have dispersed over astounding distances, on average their dispersal range is quite limited. Since most rivers on the Scandinavian peninsula flow west to east from the mountains to the Baltic Sea, those mountains should function as a barrier to gene flow. The geography of Finland and Northern Sweden with thousands of lakes that are often interconnected through wetlands as well as the shared Bothnian coastline should provide conditions that allow for gene exchange and result in higher genetic variation.

In the hierarchical clustering dendrogram (Figure 3), the Norwegian and Swedish populations had a closer relationship with each other, while the Finnish population was more distant. The DAPC (Figure 5) reinforces the genetic distances observable in the dendrogram, aligning with geography and the knowledge of a postglacial recolonization of Scandinavia from Finland.

When combining the geographical clusters with the dendrogram and the results of the PCA (Figure 4) and sPCA (Figure 6), as well as using Figure 7 and Table 2, several interesting connections become apparent.

The clear Norwegian cluster visible in both PCA and sPCA (Figure 4 and 6) is notable for it being virtually identical to sector B in the dendrogram and the Vestland geographical cluster in Western Norway. This points to the cluster having a distinct genetic identity, as has been previously identified by Mucci et al. 2010. This is likely a result of the geography and relative isolation from the metapopulation. The particularly distinct genetic profile of this group observed in Figure 3 may suggest the influence of genetic drift or the result of colonization from a different source population. It could be interesting to assess their genetic relation to otters in the Shetland Islands, which is the geographically closest population, also considering that the North Atlantic Current flows past the Shetlands towards Norway.

The Norwegian and Swedish outliers (LuN_27, 29; LuS_31) that are closest to the Finnish samples, as seen in the PCA and sPCA (Figure 4 and 6), belong to sectors C, D, and F. They are located in Lapland, i.e. close to Finland. Through previous studies using microsatellite markers (Honnen et al. 2015, Tison et al. 2015), Lapland has already been identified as a region of higher admixture, providing support for the genetic relationships found in this thesis project. In addition *LuS_31* is closest related to individuals on the Finnish Bothnian coast and on the Åland Islands (Figure 3 and Figure 7) hinting at gene flow around the Bothnian Bay. Which is further indicated by sector G (Figure 3) that shows closeness between *LuS__11* in Södermanland and three Finnish individuals, one of them from the Finnish west coast. It could also identify the Åland Islands as one of the recolonization pathways into Central and Southern Sweden.

The Norwegian individuals that are more closely related to Swedish otters seen in Figures 4 and 6, largely correspond to sector I in the dendrogram, and belong to the Trøndelag-cluster pointing to a connection between these groups. Similar findings were recorded by Mucci et al. 2010, detecting genetic closeness between these two otter populations.

The Vestføld-outlier (*LuN_08*) is located in sector J which predominantly contains Swedish individuals, both from the Götaland-cluster and the Svealand-cluster. This relationship with *LuN_08* is likely due to otters recolonizing the Scandinavian peninsula from the North, with otters in Eastern Norway descending from the same group as otters in Central and Southern Sweden, and being most closely related to the individuals *LuS_14* and *LuS_26* (Figure 3) that belong to the Svealand-cluster.

The Svealand-cluster contains individuals from sectors I and J, which might be explained by rivers flowing from west to east through the more rugged terrain in the west, and the gentler terrain around Lake Malären and in Uppland. It should be mentioned that the genetic closeness of otters from the Svealand group to otters from the Trøndelag-cluster might stem from reintroduction efforts using Norwegian otters that took place in Uppland and Södermanland (Sjöåsen 1996). Arrendal et al. 2004, did not find strong indications of effects of this reintroduction on genetic diversity using autosomal and mitochondrial microsatellite markers, but it is perhaps visible in this analysis due to usage of SNP-markers. The relation of LuF_21 to this group is potentially due to it being genetically similar to the ancestors of this sector I, as in both Figures 3 and 5 it lies on the border of the Finnish cluster.

The sector L is particularly interesting, as it suggests genetic closeness between two individuals from Finland (*LuF_12* and *LuF_19*) to individuals in Götaland (*LuS_01, 04, 08, 09, 10, 21*), Norrland (*LuS_04*), and Finnmark (*LuN_05*). The reason for this could be unique genetics or mutation.

Spatial and Landscape Genetics

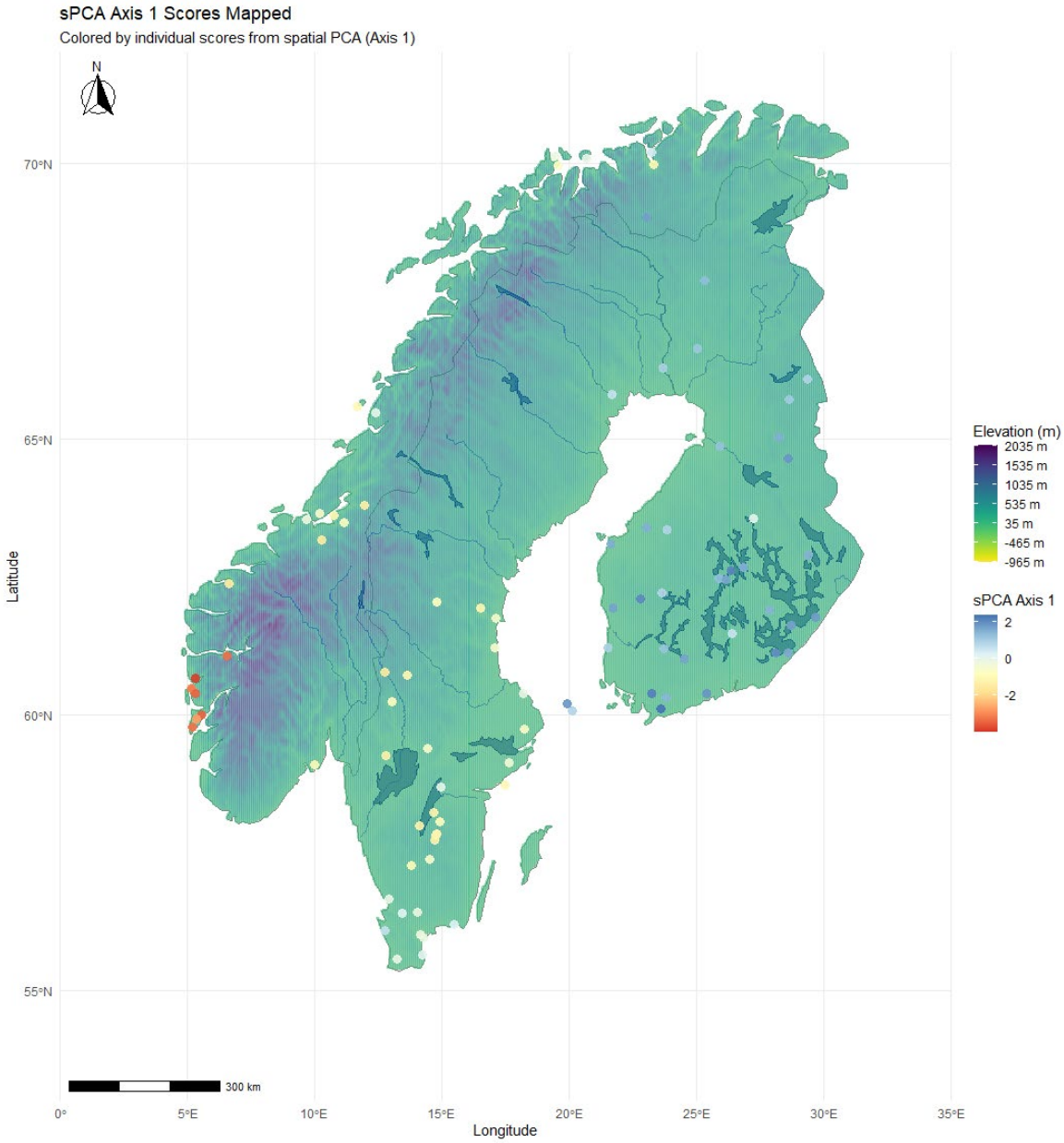

Figure 9 - sPCA Axis 1 (primary structure) scores mapped, blue and red showing respective extremes of the genetic spectrum, while light blue/yellow denotes intermediate genetics. Also showing landscape features such as elevation, rivers, and lakes.

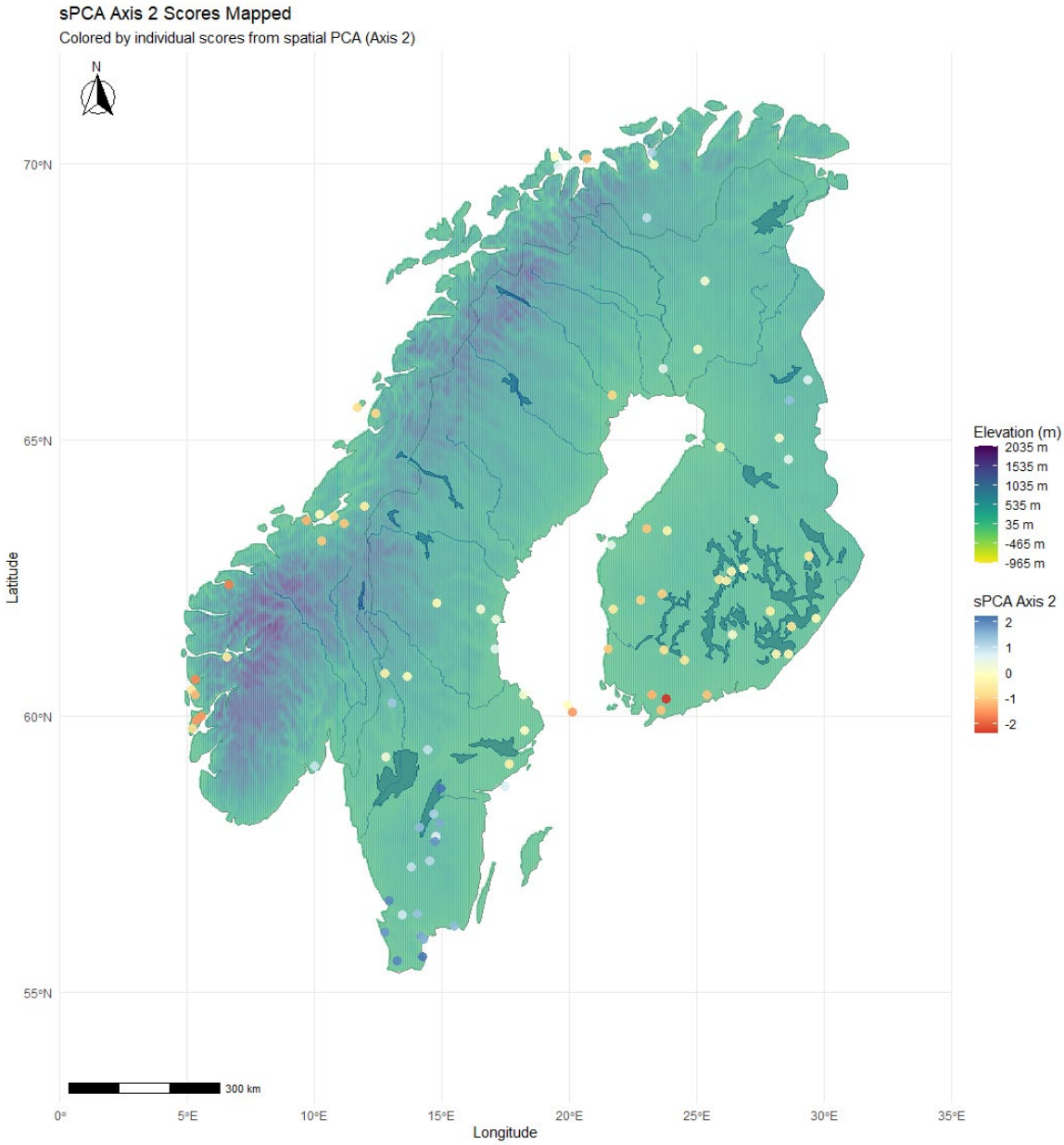

Figure 10 – sPCA Axis 2 (secondary structure) scores mapped, blue and red showing respective extremes of the genetic spectrum, while light blue/yellow denotes intermediate genetic structure. Also showing landscape features such as elevation, rivers, and lakes.

The results of the Mantel test show the presence of IBD through the moderate positive linear correlation between geographic and genetic distance. This is becoming especially apparent with the regression line in Figure 8 which suggests this correlation. In addition, by mapping the scores of the sPCA axes 1 and 2 (Figures 9 and 10) more spatial patterns become visible.

Through mapping Axis 1 in Figure 9, an East-West gradient on the genetic spectrum becomes apparent, with the Vestland-cluster and the Southern Finnish samples representing the two extremes of the spectrum, which again points to IBD presence.

By mapping Axis 2 in Figure 10 it becomes apparent that the Southern extreme of the Fennoscandian otter metapopulation exhibits localized genetic structure, showing differentiation from the rest of the sample group, further reinforcing the presence of IBD. This structuring in Southern Sweden has also been previously reported by Honnen et al. 2015, and Tison et al. 2015.

Overall, I was able to find support for parts of my hypothesis, that there is a highly distinct group of otters in West Norway (Vestland-cluster), admixture between populations is higher in Lapland and around the Bothnian bay, genetic structuring in Southern Sweden, and the presence of IBD. With this I succeeded in identifying and assessing population structure in Fennoscandian otters.

However, geographic mapping revealed a lack of sample coverage in Northern Sweden, which did not make the assessment of a possible genetic uniqueness of inland otters possible. In addition, the Norwegian samples are roughly divided into three geographical clusters with a large part of the coastline lacking samples, so that I could not fully assess these parts of my hypothesis. In total, the Finnish sampling coverage is best, however the North is more sparsely sampled. To reinforce the findings and conclusions made in this thesis project, a systematic coverage of Fennoscandia would be of great benefit, especially to thoroughly assess the intermediary genetic profile of Swedish otters. In addition, it should be noted that only data from 93 individuals and 63 SNP-loci was assessed in this thesis project, so with a larger sample size a higher resolution could be achieved. All these sampling limitations are inherent in the opportunistic sampling of deceased otters. A valuable addition to assess potential biases in this method of sampling would be the mapping of density of road networks and amount of traffic, linking this to occurrence of otters colliding with vehicles. This is beyond the scope of this thesis, but it merits further research.

Another interesting dimension to add could be the relationship of otters in southernmost Sweden to the population in Denmark, as well as otters in Western Norway to otters in the Shetland Islands. To further strengthen findings made in this thesis project, an analysis using a larger SNP-sample set and the program STRUCTURE could be conducted.

Conclusion

Eurasian otters in Fennoscandia exhibit clear indications of genetic structure and IBD, with distinct subpopulations driven by geographical features. Population structures that were previously identified are now reinforced using a different genomic marker with the benefit of being easier to standardize for future studies. Nevertheless, the otters genetic profile ought to be further studied and analysed, as it is a crucial factor to be considered in conservation actions directed at otters.

5. Acknowledgements

5.1 Giving Thanks

I want to sincerely thank my supervisor Anita Norman for the opportunity to write my thesis on this topic and for her guidance and support throughout the process of analysing and interpreting the dataset.

5.2 AI-Use during coding

In the process of analysing the data in R-Studio during this Bachelor's thesis, the Artificial Intelligence tool ChatGPT, by OpenAI, was used to identify faults and bugs in the code, streamlining, debugging, and to optimize runtime, as well as to search for potential code and colour packages.

6. References

Allendorf, F.W., Luikart, G.H. and Aitken, S.N., 2012. *Conservation and the genetics of populations*. John Wiley & Sons.

Arrendal, J., Walker, C.W., Sundqvist, A.K., Hellborg, L. and Vilà, C., 2004. Genetic evaluation of an otter translocation program. *Conservation Genetics*, *5*, pp.79-88. https://doi.org/10.1023/B:COGE.0000014059.49606.dd

Council of Europe, 1979. *Bern Convention on the Conservation of European Wildlife and Natural Habitats*. Appendix II. Council of Europe. Available at: https://rm.coe.int/168078e2ff [Accessed 31st March 2025].

Dallas, J.F., Marshall, F., Piertney, S.B., Bacon, P.J. and Racey, P.A., 2002. Spatially restricted gene flow and reduced microsatellite polymorphism in the Eurasian otter Lutra lutra in Britain. *Conservation Genetics*, *3*, pp.15-28. https://doi.org/10.1023/A:1014259218632

De Vries, A., Ripley, B.D. and de Vries, M.A., 2024. Package 'ggdendro'. https://CRAN.R-project.org/package=ggdendro

Directive, H., 1992. Council Directive 92/43/EEC of 21 May 1992 on the conservation of natural habitats and of wild fauna and flora. *Official Journal of the European Union*, *206*(7), p. 36. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:01992L0043-19920610 [Accessed 4th April 2025].

Dray, S., Dufour, A.-B. and Thioulouse, J., 2023. ade4: Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences. Version 1.7-22. [online] CRAN. Available at: https://cran.r-project.org/package=ade4 [Accessed 8 May 2025].

Galili, T., 2015. dendextend: An R package for visualizing, adjusting and comparing trees of hierarchical clustering. *Bioinformatics*, *31*(22), pp.3718-3720. https://doi.org/10.1093/bioinformatics/btv428

GeoHack (n.d.). *GeoHack*. Available at: https://www.mediawiki.org/wiki/GeoHack [Accessed 12 May 2025].

Honnen, A.C., Roos, A., Stjernberg, T. and Zachos, F.E., 2015. Genetic analysis of Eurasian otters (Lutra lutra) reveals high admixture in Finland and pronounced differentiation in Sweden. *Mammalian Biology*, *80*, pp.47-53. https://doi.org/10.1016/j.mambio.2014.09.005

Jombart, T. and Ahmed, I., 2011. adegenet 1.3-1: new tools for the analysis of genome-wide SNP data. *Bioinformatics*, *27*(21), pp.3070-3071. Available at: Link [Accessed 6th May 2024]

Kruuk, H., 2006. *Otters: ecology, behaviour and conservation*. OUP Oxford.

Leoncini, F., Semenzato, P., Di Febbraro, M., Loy, A. and Ferrari, C., 2023. Come back to stay: landscape connectivity analysis for the Eurasian otter (Lutra lutra) in the

western Alps. *Biodiversity and Conservation*, *32*(2), pp.653-669. https://doi.org/10.1007/s10531-022-02517-3

Loy, A., Kranz, A., Oleynikov, A., Roos, A., Savage, M. and Duplaix, N., 2022. Lutra lutra (amended version of 2021 assessment). *The IUCN red list of threatened species*, *2022*. https://dx.doi.org/10.2305/IUCN.UK.2022-2.RLTS.T12419A218069689.en

Manel, S., Schwartz, M.K., Luikart, G. and Taberlet, P., 2003. Landscape genetics: combining landscape ecology and population genetics. *Trends in ecology & evolution*, *18*(4), pp.189-197. https://www.fs.usda.gov/rm/pubs_other/rmrs_2003_manel_s001.pdf [Accessed 4 April 2025].

Mucci, N., Arrendal, J., Ansorge, H., Bailey, M., Bodner, M., Delibes, M., Ferrando, A., Fournier, P., Fournier, C., Godoy, J.A. and Hajkova, P., 2010. Genetic diversity and landscape genetic structure of otter (Lutra lutra) populations in Europe. *Conservation Genetics*, *11*, pp.583-599. https://doi.org/10.1007/s10592-010-0054-3

Natural Earth (2021) *Natural Earth vector and raster map data* [Dataset]. Available at: https://www.naturalearthdata.com [Accessed 6th May 2025].

Pebesma, E. and Bivand, R., 2023. *Spatial data science: With applications in R*. Chapman and Hall/CRC. https://doi.org/10.1201/9780429459016

Quaglietta, L., Fonseca, V.C., Hájková, P., Mira, A. and Boitani, L., 2013. Fine-scale population genetic structure and short-range sex-biased dispersal in a solitary carnivore, Lutra lutra. *Journal of Mammalogy*, *94*(3), pp.561-571. Available at: Link [Accessed 31st March 2025].

Ralls, K., Ballou, J.D. and Frankham, R., 2001. Inbreeding and outbreeding. *Encyclopedia of biodiversity*. Available at: https://repository.si.edu/bitstream/handle/10088/21036/nzp_2013_inbreeding_and_outbreeding.pdf [Accessed 12th May 2025].

R Core Team (2024) *R: A language and environment for statistical computing* (Version 4.4.2) [Computer software]. Available at: https://www.R-project.org/ [Accessed 6th May 2025].

Sjöåsen, T., 1996. Survivorship of captive-bred and wild-caught reintroduced European otters Lutra lutra in Sweden. *Biological Conservation*, *76*(2), pp.161-165. https://doi.org/10.1016/0006-3207(95)00097-6

Slowikowski, K., 2024. ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'. Version 0.9.5. [online] CRAN. Available at: https://cran.r-project.org/package=ggrepel [Accessed 8 May 2025].

Stanton, D.W., Hobbs, G.I., McCafferty, D.J., Chadwick, E.A., Philbey, A.W., Saccheri, I.J., Slater, F.M. and Bruford, M.W., 2014. Contrasting genetic structure of the Eurasian otter (Lutra lutra) across a latitudinal divide. *Journal of Mammalogy*, *95*(4), pp.814-823. https://doi.org/10.1644/13-MAMM-A-201

Tison, J.L., Blennow, V., Palkopoulou, E., Gustafsson, P., Roos, A. and Dalén, L., 2015. Population structure and recent temporal changes in genetic variation in Eurasian otters from Sweden. *Conservation genetics*, *16*, pp.371-384. https://doi.org/10.1007/s10592-014-0664-2

Vignal, A., Milan, D., SanCristobal, M. and Eggen, A., 2002. A review on SNP and other types of molecular markers and their use in animal genetics. *Genetics selection evolution*, *34*(3), pp.275-305. Available at: https://www.gse-journal.org/articles/gse/pdf/2002/03/g340301.pdf [Accessed 1st April 2025].

Wickham, H., 2016. Toolbox. In *ggplot2: Elegant Graphics for Data Analysis* New York Springer-Verlag. Available at: https://ggplot2.tidyverse.org [Accessed 6th May 2025].

Appendix 1

Code used for Analysis in R-Studio

```
# -------------------- Install and Load Packages --------------------
install.packages(c("adegenet", "dplyr", "ggplot2", "pegas", "poppr",
"dendextend","ggspatial", "rnaturalearth", "rnaturalearthdata", "sf", "spdep",
"ggrepel","vegan","geosphere","adespatial", "elevatr", "viridis", "raster"))
# -------------------- Load Packages (Ordered by Usage) --------------------
library(adegenet)       # Used in readRDS (genind object), autosomal filtering,
genpop conversion, DAPC
library(dplyr)          # Used throughout for data manipulation
library(ggplot2)        # Used for plots (HWE, MAF, PCA, dendrograms, etc.)
library(dendextend)     # Used to color dendrogram branches by population
library(pegas)          # Used for Hardy-Weinberg Equilibrium test
library(ggrepel)        # Used for adding text labels on PCA/sPCA and maps
library(tidyr)          # Used for allele frequency manipulation
library(stringr)        # Used for locus name filtering
library(reshape2)       # Used during allele frequency transformations
library(ggdendro)       # Used to extract dendrogram data
library(ggspatial)      # Used for map annotations (north arrow, scale)
library(sf)             # Used for spatial data (metadata, lakes, rivers)
library(rnaturalearth)  # Used for downloading map data
library(rnaturalearthdata) # Supporting data for rnaturalearth
library(vegan)          # Used for Mantel test
library(geosphere)      # Used for geographic distances (distm function)
library(spdep)          # Needed for spca() adjacency structures
library(adespatial)     # Used for sPCA
library(viridis)        # Used for elevation and map color gradients
library(elevatr)        # Used to download elevation raster
library(raster)         # Used to process elevation raster

# -------------------- Load Data --------------------
lu_data_1 <- readRDS("//storage-al.slu.se/student$/luph0001/My
Documents/OtterData/Lu_plate1.rds")
lu_meta <- read.csv("//storage-al.slu.se/student$/luph0001/My
Documents/OtterData/lu_metadata.csv", fileEncoding = "UTF-8")


# Backup
lu_safety <- lu_data_1
```

```
# -------------------- Basic QC and Summary --------------------
str(lu_data_1)
sum(is.na(lu_data_1@tab))
table(lu_data_1@pop)

otter_colors <- c("S" = "goldenrod", "N" = "red2", "F" = "blue4")

summary(lu_data_1)

missing_data_per_pop <- data.frame(
  Population = names(table(lu_data_1@pop)),
  Missing_Values = tapply(rowSums(is.na(lu_data_1@tab)), lu_data_1@pop,
sum)
)

missing_data_per_individual <- data.frame(
  Individual = rownames(lu_data_1@tab),
  Missing_Values = rowSums(is.na(lu_data_1@tab))
)
# Specify the individual to remove
individual_to_remove <- "LuF_01"

# Filter the genind object to exclude that individual
lu_data <- lu_data_1[!(rownames(lu_data_1@tab) %in% individual_to_remove),
]
rownames(lu_data@tab)
summary(lu_data)
# -------------------- Define Autosomal Loci --------------------
# Extract all locus names
all_loci <- locNames(lu_data)

# Keep only autosomal loci: exclude those with 'mt', 'x', or 'y'
auto_loci <- all_loci[!grepl("mt|x|y", all_loci, ignore.case = TRUE)]

# Subset the genind object to retain only autosomal loci
lu_auto_g <- lu_data[loc = auto_loci]
lu_auto <- as.data.frame(lu_auto_g)
lu_auto$ind <- indNames(lu_auto_g)
lu_auto$pop <- pop(lu_auto_g)
```

```
#--------------Checking Heterozygosity----------------------------------------
# Function to compute observed heterozygosity for an individual
calc_ho <- function(tab_row) {
  # Count number of loci with heterozygous genotype
  # Each locus has two alleles (coded as 0, 1, or 2)
  loci_matrix <- matrix(tab_row, ncol = 2, byrow = TRUE)
  heterozygous <- rowSums(loci_matrix == 1) == 2
  ho <- mean(heterozygous, na.rm = TRUE)
  return(ho)
}

# Calculate Ho for all individuals
het_values <- apply(lu_auto_g@tab, 1, calc_ho)

# Create a data frame for plotting
heterozygosity_df <- data.frame(
  Individual = rownames(lu_auto_g@tab),
  Heterozygosity = het_values,
  Population = pop(lu_auto_g)
)
ggplot(heterozygosity_df, aes(x = reorder(Individual, -Heterozygosity), y =
Heterozygosity, fill = Population)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = otter_colors) +
  labs(title = "Observed Heterozygosity per Individual",
       x = "Individual",
       y = "Observed Heterozygosity (Ho)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# -------------------- Genpop Conversion and Genetic Distance--------------------
lu_auto_gp <- genind2genpop(lu_auto_g)

genetic_dist <- dist.genpop(lu_auto_gp)
print(genetic_dist)
genetic_dist <- as.dist(genetic_dist)
pairDistPlot(lu_auto_g, within=FALSE, sep="-", data=TRUE,
             violin=TRUE, boxplot=TRUE, jitter=TRUE)


hc <- hclust(genetic_dist)
```

```
plot(hc, main="Genetic distances between populations")

# -------------------- HWE Test --------------------
# Perform the Hardy-Weinberg equilibrium test
hwe_result <- hw.test(lu_auto_g, B = 999)

# -------------------- Extract p-values --------------------
# Get p-values from the result
hwe_pvals <- hwe_result[, "Pr(chi^2 >)"]  # Or use "Pr.exact" for exact p-values
if you prefer

# Create a data frame for the HWE results
hwe_df <- data.frame(
  Locus = rownames(hwe_result),
  p_value = hwe_pvals
)
hwe_df$Significant <- hwe_df$p_value < 0.001
summary(hwe_df)
#Assessing significant loci and subsequent exclusion from the dataset
significant_loci <- hwe_df$Locus[hwe_df$Significant]
significant_loci
clean_loci <- setdiff(auto_loci, significant_loci)
lu_auto_g <- lu_auto_g[loc = clean_loci]

# -------------------- HWE Visualization --------------------
# Calculate percentage of significant loci
sig_count <- sum(hwe_df$Significant)
total_loci <- nrow(hwe_df)
sig_percent <- round((sig_count / total_loci) * 100, 1)

# Plot with percentage in subtitle
ggplot(hwe_df, aes(x = reorder(Locus, -p_value), y = p_value, fill = Significant))
+
  geom_bar(stat = "identity", alpha = 0.9) +
  scale_fill_manual(values = c("TRUE" = "firebrick", "FALSE" = "steelblue")) +
  geom_hline(yintercept = 0.001, linetype = "dashed") +
  coord_flip() +
  theme_minimal() +
  labs(
    title = "HWE Test: Significant Loci in Red",
```

```
    subtitle = paste0(sig_percent, "% of loci (", sig_count, "/", total_loci, ") have p
< 0.001"),
    x = "Locus", y = "p-value", fill = "Significant"
  ) +
  theme(axis.text.y = element_text(size = 6))
# -------------------- Allele Frequency & Filtering --------------------
# Calculate allele frequencies from genotype matrix (0/1/2 assumed for diploids)
allele_freqs <- as.data.frame(lu_auto_g@tab) %>%
  mutate(Population = lu_auto_g@pop) %>%
  group_by(Population) %>%
  summarise(across(everything(), ~ mean(. , na.rm = TRUE) / 2)) %>%  # divide
by 2 to scale to [0,1]
  pivot_longer(-Population, names_to = "Locus", values_to = "Allele_Frequency")
%>%
  mutate(
    Category = ifelse(str_detect(Locus, "[a-z]+"),
                      str_replace(Locus, ".*_(\\d+)([a-zA-Z]+).*", "\\2"),
                      "a")
  )

# Filter out non-autosomal loci (e.g., mtDNA, sex chromosomes)
lu_allele <- allele_freqs %>%
  filter(!str_detect(Category, regex("mt|x|y", ignore_case = TRUE)))

# -------------------- MAF --------------------

# Safe MAF calculation — filters invalid values
calculate_maf <- function(frequencies) {
  frequencies <- ifelse(is.na(frequencies) | frequencies < 0 | frequencies > 1, NA,
frequencies)
  pmin(frequencies, 1 - frequencies)
}
# Apply MAF function
lu_allele <- lu_allele %>%
  mutate(MAF = calculate_maf(Allele_Frequency))
# -------------------- MAF Summary --------------------

# Summary stats by population
maf_by_population_autosomal <- lu_allele %>%
  filter(!is.na(MAF)) %>%  # remove any MAFs that are still NA
  group_by(Population) %>%
```

```
  summarise(
    Avg_MAF = mean(MAF, na.rm = TRUE),
    Median_MAF = median(MAF, na.rm = TRUE),
    Min_MAF = min(MAF, na.rm = TRUE),
    Max_MAF = max(MAF, na.rm = TRUE)
  )
# -------------------- MAF Plot --------------------

# Plot MAFs across loci and populations
ggplot(lu_allele %>% filter(!is.na(MAF)), aes(x = Locus, y = MAF, fill =
Population)) +
  geom_bar(stat = "identity", position = "dodge", alpha = 0.7) +
  theme_minimal() +
  labs(title = "MAF by Locus and Population",
       x = "Locus (SNP)", y = "Minor Allele Frequency (MAF)") +
  scale_fill_manual(values = otter_colors) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Filter for monomorphic loci
monomorphic_loci <- c("Lu_141", "Lu_165")
relevant_loci <- setdiff(auto_loci, monomorphic_loci)
lu_allele <- lu_allele %>%
  mutate(Locus_Base = sub("\\..*", "", Locus))  # remove everything after the first
dot
lu_allele <- lu_allele %>% filter(Locus_Base %in% relevant_loci)
lu_auto_g <- lu_auto_g[loc = relevant_loci]

# Apply MAF function
lu_allele <- lu_allele %>%
  mutate(MAF = calculate_maf(Allele_Frequency))
# -------------------- MAF Summary --------------------

# Summary stats by population
maf_by_population_autosomal <- lu_allele %>%
  filter(!is.na(MAF)) %>%  # remove any MAFs that are still NA
  group_by(Population) %>%
  summarise(
    Avg_MAF = mean(MAF, na.rm = TRUE),
    Median_MAF = median(MAF, na.rm = TRUE),
    Min_MAF = min(MAF, na.rm = TRUE),
    Max_MAF = max(MAF, na.rm = TRUE)
```

```
  )

# ---------- MAF re-Plot ----------

# Plot MAFs across loci and populations
ggplot(lu_allele %>% filter(!is.na(MAF)), aes(x = Locus, y = MAF, fill =
Population)) +
  geom_bar(stat = "identity", position = "dodge", alpha = 0.7) +
  theme_minimal() +
  labs(title = "MAF by Locus and Population",
       x = "Locus (SNP)", y = "Minor Allele Frequency (MAF)") +
  scale_fill_manual(values = otter_colors) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# ---------- Allele Frequency Plot ----------
ggplot(lu_allele, aes(x = Locus, y = Allele_Frequency, fill = Population)) +
  geom_bar(stat = "identity", position = "dodge", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Allele Frequencies by Locus and Population",
       x = "Locus (SNP)", y = "Allele Frequency") +
  scale_fill_manual(values = otter_colors) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
# ---------- Merge Metadata ----------
lu_meta$Individual <- as.character(lu_meta$Individual)
lu_meta <- lu_meta %>% filter(Individual %in% indNames(lu_auto_g))
metadata_df <- lu_meta %>% dplyr::select(Individual, Population_Meta =
Population, Latitude, Longitude, Location)

# ---------- PCA ----------
gen_matrix_autosomal <- scaleGen(lu_auto_g, NA.method = "mean")

pca_result_autosomal <- dudi.pca(gen_matrix_autosomal, center = TRUE, scale =
TRUE, scannf = FALSE, nf = 3)

pca_scores_autosomal <- as.data.frame(pca_result_autosomal$li) %>%
  mutate(Individual = indNames(lu_auto_g),
         Population = lu_auto_g@pop) %>%
  left_join(metadata_df, by = "Individual")

ggplot(pca_scores_autosomal, aes(x = Axis1, y = Axis2, color = Population, label
= Individual)) +
```

```
  geom_point(size = 3) +
  geom_text_repel(size = 2.5, max.overlaps = 20) +
  theme_minimal() +
  labs(title = "PCA (Autosomal SNPs)",
       x = paste0("PC1 (",
round(pca_result_autosomal$eig[1]/sum(pca_result_autosomal$eig)*100, 2),
"%)"),
       y = paste0("PC2 (",
round(pca_result_autosomal$eig[2]/sum(pca_result_autosomal$eig)*100, 2),
"%)")) +
  scale_color_manual(values = otter_colors)

# ---------- DAPC ----------
dapc_result_autosomal <- dapc(gen_matrix_autosomal, grp = lu_auto_g@pop,
n.pca = 50, n.da = 2)

scatter(dapc_result_autosomal, col = c("blue4", "red2", "goldenrod"), main =
"DAPC - Autosomal SNPs")

eig_values <- dapc_result_autosomal$eig
ggplot(data.frame(LD = seq_along(eig_values), Eigenvalue = eig_values), aes(x =
LD, y = Eigenvalue)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "DAPC - Eigenvalues", x = "Linear Discriminants (LD)", y =
"Eigenvalue") +
  theme_minimal()

# ---------- Dendrogram ----------
gen_dist_matrix_autosomal <- dist(gen_matrix_autosomal)
hc_autosomal <- hclust(gen_dist_matrix_autosomal)

dend <- as.dendrogram(hc_autosomal)
dend_order <- order.dendrogram(dend)
names_ordered <- indNames(lu_auto_g)[dend_order]
pop_ordered <- as.character(pop(lu_auto_g))[dend_order]

labels(dend) <- names_ordered
dend <- dend %>%
  set("branches_col", c("S" = "goldenrod", "N" = "red2", "F" =
"blue4")[pop_ordered])
```

```
dend_data <- dendro_data(dend, type = "rectangle")
dend_data$labels$pop <- pop_ordered

ggplot() +
  geom_segment(data = dend_data$segments,
               aes(x = x, y = y, xend = xend, yend = yend),
               color = "gray30") +
  geom_text(data = dend_data$labels,
            aes(x = x, y = y - 0.02 * max(dend_data$segments$y), label = label, color
= pop),
            angle = 90, hjust = 1, size = 2.8) +
  scale_color_manual(values = otter_colors) +
  labs(title = "Dendrogram of Autosomal SNPs",
       x = NULL, y = "Genetic Distance", color = "Population") +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    panel.grid = element_blank(),
    legend.position = "top"
  )
# -------------------- Clustering for Box Labels --------------------

# Number of clusters
k <- 12
clusters <- cutree(hc_autosomal, k = k)

# Create cluster info table aligned with dendrogram order
cluster_df <- tibble(label = names_ordered,
                     cluster = clusters[dend_order])

# Add x position for each label
dend_data <- dendro_data(dend, type = "rectangle")
cluster_df$x <- dend_data$labels$x

# Assign letters A–L to clusters by order of appearance
cluster_labels <- cluster_df %>%
  group_by(cluster) %>%
  summarize(x = mean(x)) %>%
  arrange(x) %>%
  mutate(letter = LETTERS[1:n()])
```

```
# Cluster boundary positions, shifted right to avoid overlap
cluster_bounds <- cluster_df %>%
  group_by(cluster) %>%
  summarize(x = max(x)) %>%
  mutate(x_shifted = x + 0.6)

# -------------------- Plot --------------------

ggplot() +
  # Dendrogram segments
  geom_segment(data = dend_data$segments,
               aes(x = x, y = y, xend = xend, yend = yend),
               color = "gray30") +

  # Tip labels colored by population
  geom_text(data = dend_data$labels,
            aes(x = x, y = y - 0.02 * max(dend_data$segments$y),
                label = label, color = pop_ordered),
            angle = 90, hjust = 1, size = 2.8) +

  # Color scale for populations
  scale_color_manual(values = otter_colors) +

  # Dashed, shifted vertical lines for cluster boundaries
  geom_vline(data = cluster_bounds,
             aes(xintercept = x_shifted),
             linetype = "dashed", color = "black") +

  # Cluster group labels (A–L)
  geom_text(data = cluster_labels,
            aes(x = x + 0.3, y = max(dend_data$segments$y) + 1, label = letter),
            size = 4, fontface = "bold") +

  # Labels and theme
  labs(title = "Dendrogram of Autosomal SNPs",
       x = NULL, y = "Genetic Distance", color = "Population") +
  theme_minimal(base_size = 12) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
```

```
    panel.grid = element_blank(),
    legend.position = "top"
  )
# -------------------- Map Visualization --------------------

# --- PREP ---
label_by_individual <- "Individual"  # or "Location"
label_by_location <- "Location"      # Label using "Location"

# Clean your data
lu_meta_clean <- lu_meta %>%
  filter(!is.na(Latitude) & !is.na(Longitude))

# Spatial conversion
lu_meta_sf <- st_as_sf(lu_meta_clean, coords = c("Longitude", "Latitude"), crs =
4326)

# Country filter
countries_of_interest <- c("Norway", "Finland", "Sweden")
world <- ne_countries(scale = "medium", returnclass = "sf")
map_data <- world %>% filter(name %in% countries_of_interest)
map_union <- st_union(map_data)

# Caching paths for lakes, rivers, and elevation data
lakes_file <- "lakes_data.rds"
rivers_file <- "rivers_data.rds"
elevation_file <- "elevation_data.rds"

# Check if cached lakes data exists, otherwise download and save
if (file.exists(lakes_file)) {
  lakes <- readRDS(lakes_file)
} else {
  lakes <- ne_download(scale = 50, type = "lakes", category = "physical",
returnclass = "sf")
  lakes <- st_make_valid(lakes)
  saveRDS(lakes, lakes_file)
}

# Check if cached rivers data exists, otherwise download and save
if (file.exists(rivers_file)) {
  rivers <- readRDS(rivers_file)
```

```
} else {
  rivers <- ne_download(scale = 10, type = "rivers_lake_centerlines", category =
"physical", returnclass = "sf")
  rivers <- st_make_valid(rivers)
  saveRDS(rivers, rivers_file)
}

# Crop lakes and rivers to the map extent for performance
sf_use_s2(FALSE)  # Disable s2 to avoid geometry issues
lakes_crop <- st_intersection(st_crop(lakes, st_bbox(map_data)), map_union)
rivers_crop <- st_intersection(st_crop(rivers, st_bbox(map_data)), map_union)
sf_use_s2(TRUE)  # Re-enable s2

# --- ELEVATION DATA ---
# Check if cached elevation raster exists, otherwise compute and save
if (file.exists(elevation_file)) {
  elevation <- readRDS(elevation_file)
} else {
  elevation <- get_elev_raster(locations = map_data, z = 4, clip = "location")
  saveRDS(elevation, elevation_file)
}

# Convert elevation raster to data frame for ggplot
elev_df <- as.data.frame(rasterToPoints(elevation))
colnames(elev_df) <- c("Longitude", "Latitude", "Elevation")

# --- LABEL POINTS ---
# Label points for "Individual"
label_points_individual <- lu_meta_clean %>%
  group_by_at(c(label_by_individual, "Population")) %>%
  summarise(
    Longitude = mean(Longitude, na.rm = TRUE),
    Latitude = mean(Latitude, na.rm = TRUE),
    Label = first(!!sym(label_by_individual)),
    .groups = "drop"
  )

# Label points for "Location"
label_points_location <- lu_meta_clean %>%
  group_by_at(c(label_by_location, "Population")) %>%
  summarise(
```

```
    Longitude = mean(Longitude, na.rm = TRUE),
    Latitude = mean(Latitude, na.rm = TRUE),
    Label = first(!!sym(label_by_location)),
    .groups = "drop"
  )

# --- COLOR PALETTE ---
otter_colors <- c("S" = "goldenrod", "N" = "red2", "F" = "blue4")

# --- PLOT 1: Individual Labels ---
p1 <- ggplot() +
  # Countries (base map)
  geom_sf(data = map_data, fill = "gray90", color = "gray50") +

  # Water features
  geom_sf(data = lakes_crop, fill = "darkblue", color = NA, alpha = 0.6) +
  geom_sf(data = rivers_crop, color = "blue3", size = 0.6, alpha = 0.8) +

  # Elevation (background)
  geom_tile(data = elev_df, aes(x = Longitude, y = Latitude, fill = Elevation),
alpha = 0.5) +
  scale_fill_viridis(option = "D", direction = -1, name = "Elevation (m)",
                     limits = c(min(elev_df$Elevation), max(elev_df$Elevation)),
                     breaks = seq(min(elev_df$Elevation), max(elev_df$Elevation), by =
500),
                     labels = function(x) paste0(x, " m")) +

  # Otter points
  geom_jitter(data = lu_meta_clean,
              aes(x = Longitude, y = Latitude, color = Population),
              width = 0.1, height = 0.1, size = 3, alpha = 0.9) +

  # Labels (on top of everything)
  geom_text_repel(data = label_points_individual,
                  aes(x = Longitude, y = Latitude, label = Label),
                  size = 3.5, fontface = "italic", box.padding = 0.3, max.overlaps = Inf)
+

  # North arrow & scale bar
  annotation_scale(location = "bl", width_hint = 0.2) +
```

```
  annotation_north_arrow(location = "tl", which_north = "true", style =
north_arrow_fancy_orienteering) +

  # Custom colors, coordinate focus, theme
  scale_color_manual(values = otter_colors) +
  coord_sf(xlim = c(0, 35), ylim = c(53, 72), expand = FALSE) +
  theme_minimal() +
  labs(
    title = paste("Geographic Distribution of Otter Samples (", label_by_individual,
")", sep = ""),
    x = "Longitude", y = "Latitude", color = "Population"
  )

# --- PLOT 2: Location Labels ---
p2 <- ggplot() +
  # Countries (base map)
  geom_sf(data = map_data, fill = "gray90", color = "gray50") +

  # Water features
  geom_sf(data = lakes_crop, fill = "darkblue", color = NA, alpha = 0.6) +
  geom_sf(data = rivers_crop, color = "blue3", size = 0.6, alpha = 0.8) +

  # Elevation (background)
  geom_tile(data = elev_df, aes(x = Longitude, y = Latitude, fill = Elevation),
alpha = 0.5) +
  scale_fill_viridis(option = "D", direction = -1, name = "Elevation (m)",
                     limits = c(min(elev_df$Elevation), max(elev_df$Elevation)),
                     breaks = seq(min(elev_df$Elevation), max(elev_df$Elevation), by =
500),
                     labels = function(x) paste0(x, " m")) +

  # Otter points
  geom_jitter(data = lu_meta_clean,
              aes(x = Longitude, y = Latitude, color = Population),
              width = 0.1, height = 0.1, size = 3, alpha = 0.9) +

  # Labels (on top of everything)
  geom_text_repel(data = label_points_location,
                  aes(x = Longitude, y = Latitude, label = Label),
                  size = 3.5, fontface = "italic", box.padding = 0.3, max.overlaps = Inf)
+
```

```
  # North arrow & scale bar
  annotation_scale(location = "bl", width_hint = 0.2) +
  annotation_north_arrow(location = "tl", which_north = "true", style =
north_arrow_fancy_orienteering) +

  # Custom colors, coordinate focus, theme
  scale_color_manual(values = otter_colors) +
  coord_sf(xlim = c(0, 35), ylim = c(53, 72), expand = FALSE) +
  theme_minimal() +
  labs(
    title = paste("Geographic Distribution of Otter Samples (", label_by_location,
")", sep = ""),
    x = "Longitude", y = "Latitude", color = "Population"
  )

# Print the plots
print(p1)
print(p2)

# Save to high-res PNG
ggsave(p1)
ggsave(p2)
# -------------------- Mantel Test --------------------
# Arrange metadata so it matches the order of individuals in the genetic data
coords <- metadata_df %>%
  arrange(match(Individual, indNames(lu_auto_g))) %>%
  dplyr::select(Longitude, Latitude)

# Compute geographic distance matrix (in kilometers)
geo_mat <- distm(coords, fun = distHaversine) / 1000  # Returns a full matrix

# Assign proper row and column names from the matching individual names
ind_names <- indNames(lu_auto_g)
rownames(geo_mat) <- ind_names
colnames(geo_mat) <- ind_names

# Convert to 'dist' object
geo_dist <- as.dist(geo_mat)
```

```
# Compute genetic distance (make sure gen_matrix_autosomal matches lu_auto_g
order)
gen_dist <- dist(gen_matrix_autosomal)

# Run Mantel test
mantel_result <- vegan::mantel(gen_dist, geo_dist, method = "pearson",
permutations = 999)

# View results
print(mantel_result)

# Convert distance matrices to numeric vectors
gen_vec <- as.vector(gen_dist)
geo_vec <- as.vector(geo_dist)

# Now plot the relationship
plot(geo_vec, gen_vec,
     main = "Genetic vs Geographic Distance",
     xlab = "Geographic Distance (km)",
     ylab = "Genetic Distance",
     pch = 20, col = "steelblue")

# Add linear regression line
abline(lm(gen_vec ~ geo_vec), col = "firebrick", lwd = 2)

# -------------------- Prepare Coordinates --------------------
# Match metadata to genind individuals
coords <- metadata_df %>%
  filter(Individual %in% indNames(lu_auto_g)) %>%
  arrange(match(Individual, indNames(lu_auto_g))) %>%
  dplyr::select(Longitude, Latitude)

coords_jittered <- jitter(as.matrix(coords), amount = 0.0005)

# -------------------- Run sPCA --------------------
spca_result <- spca(obj = lu_auto_g,
                    xy = coords_jittered,
                    type = 1,       # Delaunay triangulation
                    scannf = FALSE,
                    nfposi = 2,     # Number of global components
                    nfnega = 0)     # No local components here
```

```
# -------------------- Plot Eigenvalues --------------------
barplot(spca_result$eig,
        main = "sPCA Eigenvalues",
        col = "steelblue",
        border = NA,
        ylab = "Eigenvalue",
        xlab = "sPCA Axis",
        names.arg = seq_along(spca_result$eig),
        las = 1)

# -------------------- Prepare Data for Plotting --------------------
# Create sPCA scores data frame with SampleID
spca_scores <- as.data.frame(spca_result$li)
spca_scores$SampleID <- rownames(spca_scores)

# Make sure SampleID is correct in metadata (already fixed)
metadata_df$SampleID <- metadata_df$Individual  # This must match rownames
of spca_scores

# Create Population data frame
population_df <- data.frame(
  SampleID = indNames(lu_auto_g),
  Population = as.character(lu_auto_g@pop)
)

# Merge spca_scores with population and location info
spca_data <- spca_scores %>%
  left_join(population_df, by = "SampleID")

# Update Location column from metadata
spca_data$Location <- metadata_df$Location[match(spca_data$SampleID,
metadata_df$SampleID)]

# -------------------- sPCA Score Scatterplot (with ggplot) --------------------
# Define population colors
otter_colors <- c("S" = "goldenrod", "N" = "red2", "F" = "blue4")

# Plot first two sPCA axes with labels
ggplot(spca_data, aes(x = `Axis 1`, y = `Axis 2`, color = Population, label =
SampleID)) +
```

```
  geom_point(aes(fill = Population), shape = 21, size = 3, alpha = 0.7) +
  geom_text_repel(size = 3, box.padding = 0.3, max.overlaps = Inf) +
  scale_color_manual(values = otter_colors) +
  scale_fill_manual(values = otter_colors) +
  labs(
    title = "sPCA - Individual Scores by Population",
    x = "sPCA Axis 1",
    y = "sPCA Axis 2",
    color = "Population"
  ) +
  theme_minimal() +
  theme(
    legend.position = "top",
    plot.title = element_text(hjust = 0.5)
  )

# ---------- Spatial Network + sPCA Scores ----------
# Get sPCA axis scores
spca_df <- data.frame(
  Longitude = coords_jittered[, 1],
  Latitude = coords_jittered[, 2],
  Axis1 = spca_result$li[, 1],
  Axis2 = spca_result$li[, 2],
  Population = lu_auto_g@pop,
  Individual = indNames(lu_auto_g)
)

# ---------- Plot sPCA Axis 1 Spatially ----------
ggplot(spca_df, aes(x = Longitude, y = Latitude)) +
  geom_point(aes(color = Axis1), size = 4) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0)
+
  theme_minimal() +
  labs(
    title = "sPCA Axis 1 Spatial Distribution",
    color = "sPCA Axis 1 Score",
    x = "Longitude", y = "Latitude"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    legend.position = "right"
```

```
  )

# -------------------- Plot sPCA Axis 2 Spatially --------------------
ggplot(spca_df, aes(x = Longitude, y = Latitude)) +
  geom_point(aes(color = Axis2), size = 4) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0)
+
  theme_minimal() +
  labs(
    title = "sPCA Axis 2 Spatial Distribution",
    color = "sPCA Axis 2 Score",
    x = "Longitude", y = "Latitude"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    legend.position = "right"
  )

# -------------------- Define File Paths for Caching --------------------
lakes_cache <- "lakes_data.gpkg"
rivers_cache <- "rivers_data.gpkg"
world_cache <- "world_map_data.gpkg"

# -------------------- Check and Load Cached Data --------------------
# Load world map data (Norway, Finland, Sweden)
if (file.exists(world_cache)) {
  world <- st_read(world_cache)
} else {
  world <- ne_countries(scale = "medium", returnclass = "sf")
  # Save the world map to cache
  st_write(world, world_cache)
}

map_data <- world[world$name %in% c("Norway", "Finland", "Sweden"), ]

# Load lakes data
if (file.exists(lakes_cache)) {
  lakes <- st_read(lakes_cache)
} else {
  lakes <- ne_download(scale = 50, type = "lakes", category = "physical",
returnclass = "sf")
```

```
  lakes <- st_make_valid(lakes)
  # Save the lakes data to cache
  st_write(lakes, lakes_cache)
}

# Load rivers data
if (file.exists(rivers_cache)) {
  rivers <- st_read(rivers_cache)
} else {
  rivers <- ne_download(scale = 10, type = "rivers_lake_centerlines", category =
"physical", returnclass = "sf")
  rivers <- st_make_valid(rivers)
  # Save the rivers data to cache
  st_write(rivers, rivers_cache)
}
# Load elevation data
if (file.exists(elevation_file)) {
  elevation <- readRDS(elevation_file)
} else {
  elevation <- get_elev_raster(locations = map_data, z = 4, clip = "location")
  saveRDS(elevation, elevation_file)
}

# ---------- Spatial Operations ----------
sf_use_s2(FALSE)
map_union <- st_union(map_data)
lakes_crop <- st_intersection(lakes, map_union)
rivers_crop <- st_intersection(st_crop(rivers, st_bbox(map_data)), map_union)
sf_use_s2(TRUE)

# ---------- Extract sPCA Axis 1 Scores ----------
spca_scores_axis1 <- data.frame(
  Individual = indNames(lu_auto_g),
  sPCA_Axis1 = spca_result$li[, 1]
)

# Merge scores with metadata
lu_meta_map_axis1 <- lu_meta %>%
  filter(Individual %in% spca_scores_axis1$Individual) %>%
  left_join(spca_scores_axis1, by = "Individual") %>%
  filter(!is.na(Longitude) & !is.na(Latitude))
```

```
# Convert to sf object
lu_meta_sf_axis1 <- st_as_sf(lu_meta_map_axis1, coords = c("Longitude",
"Latitude"), crs = 4326)

# ---------- Plot sPCA Axis 1 on the Map ----------
map_axis1 <- ggplot() +
  # Base map
  geom_sf(data = map_data, fill = "gray90", color = "gray50") +

  # Water features
  geom_sf(data = lakes_crop, fill = "darkblue", color = "darkblue", alpha = 0.6) +
  geom_sf(data = rivers_crop, color = "blue3", size = 0.5, alpha = 0.6) +

  # Elevation (background)
  geom_tile(data = elev_df, aes(x = Longitude, y = Latitude, fill = Elevation),
alpha = 0.5) +
  scale_fill_viridis(option = "D", direction = -1, name = "Elevation (m)",
                     limits = c(min(elev_df$Elevation), max(elev_df$Elevation)),
                     breaks = seq(min(elev_df$Elevation), max(elev_df$Elevation), by =
500),
                     labels = function(x) paste0(x, " m")) +

  # Plot sPCA Axis 1
  geom_sf(data = lu_meta_sf_axis1, aes(color = sPCA_Axis1), size = 3, alpha =
0.9) +
  scale_color_distiller(palette = "RdYlBu", direction = 1, name = "sPCA Axis 1")
+

  # North arrow & scale bar
  annotation_scale(location = "bl", width_hint = 0.2) +
  annotation_north_arrow(location = "tl", which_north = "true", style =
north_arrow_fancy_orienteering) +

  # Coordinate focus and theme
  coord_sf(xlim = c(0, 35), ylim = c(53, 72), expand = FALSE) +
  theme_minimal() +

  # Labels
  labs(
    title = "sPCA Axis 1 Scores Mapped",
```

```
    subtitle = "Colored by individual scores from spatial PCA (Axis 1)",
    x = "Longitude", y = "Latitude"
  )
print(map_axis1)
# -------------------- Extract sPCA Axis 2 Scores --------------------
spca_scores_axis2 <- data.frame(
  Individual = indNames(lu_auto_g),
  sPCA_Axis2 = spca_result$li[, 2]
)

# Merge scores with metadata
lu_meta_map_axis2 <- lu_meta %>%
  filter(Individual %in% spca_scores_axis2$Individual) %>%
  left_join(spca_scores_axis2, by = "Individual") %>%
  filter(!is.na(Longitude) & !is.na(Latitude))

# Convert to sf object
lu_meta_sf_axis2 <- st_as_sf(lu_meta_map_axis2, coords = c("Longitude",
"Latitude"), crs = 4326)

# -------------------- Plot sPCA Axis 2 on the Map --------------------
map_axis2 <- ggplot() +
  # Base map
  geom_sf(data = map_data, fill = "gray90", color = "gray50") +

  # Water features
  geom_sf(data = lakes_crop, fill = "darkblue", color = "darkblue", alpha = 0.6) +
  geom_sf(data = rivers_crop, color = "blue3", size = 0.5, alpha = 0.6) +

  # Elevation (background)
  geom_tile(data = elev_df, aes(x = Longitude, y = Latitude, fill = Elevation),
alpha = 0.5) +
  scale_fill_viridis(option = "D", direction = -1, name = "Elevation (m)",
                     limits = c(min(elev_df$Elevation), max(elev_df$Elevation)),
                     breaks = seq(min(elev_df$Elevation), max(elev_df$Elevation), by =
500),
                     labels = function(x) paste0(x, " m")) +

  # Plot sPCA Axis 2
  geom_sf(data = lu_meta_sf_axis2, aes(color = sPCA_Axis2), size = 3, alpha =
0.9) +
```

```
  scale_color_distiller(palette = "RdYlBu", direction = 1, name = "sPCA Axis 2")
+

  # North arrow & scale bar
  annotation_scale(location = "bl", width_hint = 0.2) +
  annotation_north_arrow(location = "tl", which_north = "true", style =
north_arrow_fancy_orienteering) +

  # Coordinate focus and theme
  coord_sf(xlim = c(0, 35), ylim = c(53, 72), expand = FALSE) +
  theme_minimal() +

  # Labels
  labs(
    title = "sPCA Axis 2 Scores Mapped",
    subtitle = "Colored by individual scores from spatial PCA (Axis 2)",
    x = "Longitude", y = "Latitude"
  )
print(map_axis2)
```

Publishing and archiving

Approved students' theses at SLU can be published online. As a student you own the copyright to your work and in such cases, you need to approve the publication. In connection with your approval of publication, SLU will process your personal data (name) to make the work searchable on the internet. You can revoke your consent at any time by contacting the library.

Even if you choose not to publish the work or if you revoke your approval, the thesis will be archived digitally according to archive legislation.

You will find links to SLU's publication agreement and SLU's processing of personal data and your rights on this page:

- https://libanswers.slu.se/en/faq/228318

☒ YES, I, Luis Perlich, have read and agree to the agreement for publication and the personal data processing that takes place in connection with this

☐ NO, I/we do not give my/our permission to publish the full text of this work. However, the work will be uploaded for archiving and the metadata and summary will be visible and searchable.