Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Swiss Federal Institute for Forest,
Snow and Landscape Research WSL

Noninvasive genetic monitoring of mountain hare (*Lepus timidus*) individuals and distinguishing between mountain and European hares (*Lepus europaeus*)

Master Thesis

Laura Schürz

Schneehase im Schweizer National Park (Foto: Rolf Giger)

Supervisors: Dr. Felix Gugerli
Dr. Kurt Bollmann
Dr. Maik Rehnus
Prof. Dr. Rolf Holderegger

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Noninvasive genetic monitoring of mountain hare (Lepus timidus) individuals and distinguishing between mountain and European hares (Lepus europeaus)

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Schürz	Laura

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Birmensdorf, 27.03.2019	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Noninvasive genetic monitoring of mountain hare (*Lepus timidus*) individuals and distinguishing between mountain and European hares (*Lepus europaeus*)

Laura Schürz

Abstract. To be better able to estimate demographic parameters of elusive, difficult-to-spot or nocturnal animals, noninvasive genetic sampling (NGS) has been introduced. NGS allows the estimation of population genetic and demographic parameters based on molecular data obtained through the collection of animal remnants in the field. Here, NGS was applied to monitor mountain hare (*Lepus timidus*) individuals in the Swiss National Park. Additionally, the applicability of the thereby applied microsatellite markers to distinguish between European hares (*Lepus europaeus*) and mountain hares was determined. The results indicate the presence of 4.9 to 8 mountain hares per km^2 in the study area, whereby higher density estimates were obtained for spring (March/April) than fall (October). Additionally, the mean male:female sex-ratio was larger in spring (1.68) than in fall (1.11) and males showed higher probabilities for temporary migration from and into the study area. Apparent seasonal survival rates of males showed higher fluctuations than for females and annual survival estimates were larger for females than for males. Differences between mountain and European hares where shown in allele frequencies and heterozygosity values and illustrated based on a Principal Component and STRUCTURE analysis. Applying these findings to the NGS data implied a European hare to have been present in the National Park in spring 2016 at a maximum altitude of 2300m.a.s.l.

Table of Contents

Figures

Tables

INTRODUCTION

In wildlife conservation and management, population size or individual abundance, and changes in these parameters are essential to estimate population viability (Fryxell et al. 2014). These estimates can be obtained through directly counting a sample, or by indirect measurements, such as capture-mark-recapture (CMR) estimations (Fryxell et al. 2014). In classical CMR applications, animals are captured, marked with individual tags and released. Based on the ratio of marked and unmarked individuals in a second capture (recapture) occasion, individuals' densities are estimated (Fryxell et al. 2014, Mills 2013, Schwarz & Seber 1999). However, these methods require handling individuals and marking them (Silvy et al. 2012), which makes them unsuitable for the application to rare or elusive species (Jacob et al. 2010, Luikart et al. 2010).

Developments in molecular techniques have made it possible to genetically identify individuals based on remnants (e.g. feces or hair) collected in the field, for example through noninvasive genetic sampling (NGS). If genetic samples are collected at multiple points in time, demographic parameters (e.g. population size and apparent survival rates) may then be estimated using a statistical framework modified from classical CMR methods (Lukacs & Burnham 2005, Piggott & Taylor 2003). Each sample collected is genotyped at multiple molecular loci (e.g. microsatellites) and matching genotypes are assumed to originate from the same individual, whereas newly found genotypes are recorded as new individuals (Lampa et al. 2015). Since its introduction (Höss et al. 1992, Taberlet & Bouvet 1992), NGS has been applied in numerous projects, for example in wolf monitoring (*Canis lupus*, Stenglein et al. (2010)), the estimation of population abundance of grizzly (*Ursus arctos*) and black bears (*U. americanus*, Sawaya et al. (2012)) and for the estimation of contemporary dispersal and connectivity of capercaillie (*Tetrao urogallus*) in a regional population (Kormann et al. 2012). Particularly to obtain estimates of elusive, difficult-to-spot or small animals, NGS shows great potential, as individual abundance may be estimated without having to handle or even observe animals (Beja-Pereira et al. 2009, Kormann et al. 2012, Rosner et al. 2014, Waits & Paetkau 2005). However, animal remnants often contain DNA of poor quality and quantity and may consequently lead to high error rates (Creel et al. 2003, Taberlet et al. 1997, Taberlet et al. 1996, Taberlet et al. 1999, Waits & Leberg 2000). Additionally, low allelic diversity is generally expected in small populations, which increases the probability of two individuals sharing the same multilocus genotype and the probability of misidentifications (Mills et al. 2000). In noninvasive CMR methods, misidentifications of individuals may have severe consequences, such as false sibship exclusion (Wang 2004), false identification of individuals, and, thus, overestimation of population size (Creel & Rosenblatt 2013, Lukacs & Burnham 2005, Waits & Leberg 2000). To reduce errors in NGS, the multitube approach was introduced (Taberlet et al. 1996). The multitube approach instructs to replicate samples in multiple independent polymerase chain reactions (PCRs) and construct consensus genotypes based on these replicates. This method has been widely accepted to reduce and quantify error rates in NGS. However, it also highlights the importance of study-specific error estimations (Taberlet et al. 1996).

Here, NGS was applied to the mountain hare (*Lepus timidus*) as a model species using seven microsatellite markers with the goal to monitor individuals over time and assess population abundance (Rehnus & Bollmann 2016). As philopatric species showing only limited natal and breeding dispersal (Dahl & Willebrand 2005), the mountain hare is thought to be a good model species for NGS. Rehnus & Bollmann (2016) conducted a pilot study in a study area located in the Swiss National Park, comparing different sampling systems. They found DNA extraction to work best for feces not older than five days and detected a mountain hare density of 3.2 to 3.6 hares per km^2 using systematic, opportunistic and combined sampling (Rehnus & Bollmann 2016). The monitoring has been continued since spring 2014 and feces collection has been implemented during two sessions each year.

The mountain hare and the European hare (*Lepus europaeus*) are the two most widely spread species of hares in Europe. The mountain hare is an arctic/subarctic species with fragmented subpopulations in northern Europe and the Alps. The species distribution ranges from 1300 (Thulin & Flux 2003) up to 3800m.a.s.l. (Rehnus 2013). The European hare occurs throughout the lowlands of Europe and is known to occupy extensively managed arable areas (Smith & Johnston 2008a, Smith et al. 2005, Thulin 2003). In alpine areas, the European hare has its main distribution in altitudes of 500 to 1500m.a.s.l. (Bisi et al. 2015), but has been observed at up to 2300m.a.s.l. (Spitzenberger 2001). Both hare species are being hunted in Switzerland (BAFU 2018, Rehnus 2013) and are classified as "Least Concern" by the International Union for the Conservation of Nature (Smith & Johnston 2008a, 2008b). However, European hares show a negative population trend at the global level (Smith & Johnston 2008a) and in Europe (Smith et al. 2005), which is also visible in the rapid decline of hunting bags in Switzerland (BAFU 2018). For European hares in Switzerland, the main mortality causes are assumed to be habitat changes caused by agricultural intensification (Smith et al. 2005) and landscape fragmentation (Roedenbeck & Voser 2008). Furthermore, hunting and weather conditions may be additional external factors of importance for the species' decline, especially when magnified by the loss of qualitative habitats, as described above (Smith et al. 2005). For mountain hares, threats are thought to be mainly climate change (Acevedo et al. 2012), disturbances (Rehnus et al. 2014), diseases (e.g. parasites, Newey et al. (2007)) and interspecific competition with the European hare (Thulin 2003). Especially climate change can be regarded as a notable threat, as arctic and subarctic species have been identified as particularly vulnerable to changes in climatic conditions (Beever et al. 2011, Hughes 2000, Moritz et al. 2008, Parmesan 2006, Rehnus et al. 2018). Increases beyond the optimum average ambient temperature of 7 to 9°C during the reproductive season (April/May) are thought to be a main driver for the loss of suitable habitat (Rehnus et al. 2018). In contrast, temperature was found to have a positive effect on European hare population abundance (Smith et al. 2005), and the species distribution may experience an upwards shift with increasing temperature (Leach et al. 2016).

European and mountain hares have been known to hybridize in areas where species distributions overlap (Thulin 2003, Thulin et al. 2006b). Mountain hare females mate with European hare males and first generation hybrid females are fertile and may backcross with European hare males (Thulin 2003). Consecutive backcrosses of female offspring with European hare males will in a few generations result in phenotypic European hares carrying mitochondrial DNA (mtDNA) of mountain hares (Thulin et al. 2006b). As both species are morphologically similar during a large part of the year, hybrids are difficult to identify (Lönneberg 1905, Rehnus 2013). However, developments in molecular methods allow for genetic detection of hybrids through sequencing mtDNA (Thulin et al. 2006a, Thulin & Tegelström 2002) or through the application of nuclear markers such as microsatellites (Beugin et al. 2017). As mtDNA is maternally inherited, first generation *L. europaeus*-hybrids in *L. timidus* populations cannot be detected (Thulin et al. 2006b). Nevertheless, mtDNA sequencing may detect later generation hybrids or species memberships in general (Thulin et al. 2006a). To detect hybrids and assess gene flow across the species barrier, the use of nuclear markers is suggested (Beugin et al. 2017, Thulin et al. 2006b).

Here, I applied the same set of microsatellite markers as in the noninvasive genetic study to tissue samples collected by hunters, game keepers and researchers across the Alps in Europe. The goal thereby was to investigate the applicability of these markers for the identification of species and hybrids across multilocus genotypes.

In summary, the goal of this master thesis was (i) to analyze the genotyping results of the monitoring in the Swiss National Park, (ii) to investigate the applicability of the marker set to species identification and (iii) to apply the results obtained in (ii) to the noninvasively obtained molecular data.

PART I: NONINVASIVE GENETIC MONITORING

Methods

Sample collection

The study area is located in the southern part of the Swiss National Park (46°39'N, 10°11'E), spanning an area of 3.5 km^2, and was selected to represent the ecological range occupied by the mountain hare in the Swiss Alps and accessibility for sampling (Rehnus 2013). To confirm that the European hare is not present in the area, long-term observations (1979 – 2012) by park rangers and interviews with local hunters and gamekeepers from nearby hunting districts were consulted (unpublished data). The study area is in a strict nature reserve, which is inaccessible for the public in winter, but a popular area for recreational activities is summer (e.g. hiking), whereby visitors are not allowed to leave marked paths and areas. Consequently, the study area can be considered seasonally disturbance-free, which allows the study of mountain hares under presumably natural conditions (Rehnus & Bollmann 2016).

A pilot study conducted in March 2014 revealed the use of feces not older than five days and a combination of systematic and opportunistic sampling the be the most appropriate (Rehnus & Bollmann 2016). Consequently, in all subsequent years (2014 – 2018), sampling was done in an opportunistic and a systematic setup, and feces (samples) estimated to be older than five days were marked. For systematic sampling, 91 plots were placed on a 200-m square grid (Rehnus & Bollmann 2016) and opportunistic samples were collected whenever spotted between systematic sampling points. The average home range estimated by Nodari (2006) should cover 13 points of the systematic sampling grid. Thus, the sample is assumed to be of adequate resolution to detect a high proportion of the individuals present in the area (Rehnus & Bollmann 2016).

Both systematic and opportunistic sampling were conducted during a short timespan throughout two sessions in one year. During each session the population was assumed to be closed. The first sampling session was conducted in the beginning of April, which coincides with the beginning of the reproductive season of the mountain hare (Thulin 2003). Thus, in spring, only individuals that potentially contribute to the next reproductive cycle are recorded (Luikart et al. 2010). Mountain hares reproduce in March/April, and offspring are born after a gestation period of around 50 days (Pehrson & Lindlöf 1984). The second sampling session was carried out in fall each year, i.e. early October. Consequently, offspring born in the summer of the same year should be recorded in fall.

Samples were collected and stored in separate plastic tubes without touching by hand to minimize DNA contamination (Sloan et al. 2000). After collection in the field, samples were frozen and stored until analysis in the lab.

DNA extraction and amplification

To genotype mountain hare samples, the following ten nuclear microsatellite loci were used: Lsa1, Lsa2, Lsa3 (Kryger 2002), Sat2, Sat5, Sat8, Sat12 (Mougel et al. 1997), Sol30, Sol8 (Rico et al. 1994), Sol33 (Surrige et al. 1997). In addition, one marker was used to determine the sex (Sry, Wallner et al. (2001)).

DNA from feces collected in 2014 and 2015 was extracted after every sampling occasion following the protocol described by Rehnus & Bollmann (2016). DNA extraction from feces collected in 2016 – 2018 was performed with reagents from a customized sbeadex livestock kit (LGC Genomics, Berlin, Germany) on a King Fisher Flex (Thermo Fisher Scientific, USA).

1100µl LP-PVP and 1µl RNAse were added to each fecal pellet in the original 5mL sampling tube and incubated over night at room temperature. 500µl clear lysate were added to 490µl binding buffer SB and 10µl sbeadex beads in a deep well plate. If necessary, the volume of lysate was

adjusted to 500µl with LP-PVP. If particles were present in the lysate, lysate was centrifuged through a QIA shredder column for 2min at 850G. Deep well plates (LGC) were prepared for three washing steps: First wash with 400µl BN1, second wash step with 400µl TN1 and third wash step with TN2. In a 96-standard plate (LGC), 100µl of elution buffer AMP was added for each sample. Plates were transferred to the King Fisher platform and a protocol with the following conditions was run: 21min for binding step (1min at fast and 20min at normal mixing mode, including bottom mix). Beads were collected five times for 30s. For the first and the second washing step beads were released for 30s. Mixing was performed for 5min in fast and 5min in normal mixing mode, including bottom mix. Beads were collected five times for 10s. For the third washing step beads were released for 30s. Mixing was performed for 5min in fast and 5min in normal mixing mode, including bottom mix. Beads were collected five times for 30s.

The elution step was performed at 60°C. Beads were released for 30s. Mixing was performed for 10min at fast and for 10min at normal mixing mode including bottom mix. Beads were collected five times for 20s and transferred with the comb into the binding plate for disposal.

30µl of the eluted DNA was transferred to a new 96 plate for the nSSR (nuclear Simple Sequence Repeats/microsatellites) analysis and all other DNA to a 1.5ml tubes for backup and storage. DNA samples were amplified as described in Rehnus & Bollmann (2016) in three independent replicates in two multiplex PCRs, following a modified multi-tube approach (Taberlet et al. 1997, Taberlet et al. 1999). For amplification of samples from 2016 – 2018 concentration of primers were lowered to 0.2 – 0.3 uM. Fragment length analysis of samples from 2016 – 2018 was performed on an ABI3130 genetic analyzer using GeneScan LIZ 500 dye Size Standard (Thermo Fisher Scientific) and electropherograms were analyzed using GENEMAPPER v5.0 (Thermo Fisher Scientific) after each sampling session. The phenotypes of the three loci Sat2, Sat12, and Lsa2 were consistent across replicates, but could not be reliably scored as bi-allelic markers. Thus, they were not included in multilocus genotypes, but scored qualitatively using a description of phenotypic peaks (Appendix 1.5).

Genotyping

The raw genotype table output of the remaining seven loci was analyzed in R (R Core Team 2018) to find consensus genotypes for each replicated sample. For this purpose, a script was written in R (see Appendix 1.1) with the goal of simplifying future analysis. The R script considers each replicate, for which a genotype could be scored (positive PCR, Broquet & Petit (2004)), and applies the conditions listed in the following. Two positive PCRs per sample were at least deemed necessary for a consensus genotype to be scored. For three positive PCRs, a consensus genotype was defined when the replicates showed consistency in at least two PCRs. Homozygous allele combinations were only accepted if all three positive PCRs were consistent. For two positive PCRs, a consensus genotype was accepted only if both positive PCRs were consistent (see Appendix 1.2). A multilocus consensus genotype of a sample was accepted when containing not more than one missing locus to ensure a minimum number of six loci per sample. The determination of the sex was done using an assay developed by Wallner et al. (2001). The assay is only amplified in male individuals, as it amplifies part of the Y-chromosomal Sry (Wallner et al. 2001). A genotype was considered female, if none of the three replicates amplified at the Sry-locus and male if at least one of the replicates amplified (see Appendix 1.3).

The creation of consensus multilocus genotypes using replicates does not ensure error-free genotypes and it is thus essential to quantify study-specific error rates (Broquet & Petit 2004, Pompanon et al. 2005). Two technical types of errors are the most common in microsatellite genotyping (Broquet & Petit 2004): False alleles in homozygotes and allelic dropout, which is detectible as either false-homozygotes at heterozygote loci (dropout of one allele) or completely

missing replicates (dropout at two alleles). Causes of these errors may be low quantity or quality of DNA, pollution, or the presence of PCR inhibitors, particularly so in noninvasive samples (Pompanon et al. 2005).

As quantification of allelic dropout of both alleles, the amplification success rate (NA_{reps}) was calculated in R. The calculation was done based on all samples collected, including samples from around the study area, and samples which had to be discarded due to too many missing loci in the consensus multilocus genotype. NA_{reps} was calculated separately for each sampling session as the proportion of missing replicates per marker divided by the total amount of replicates for that marker. Missing replicates were defined as missing genotypes, not missing alleles, as single missing alleles were detected as false-homozygote replicates in heterozygous genotypes, which were considered separately, or missing genotypes in homozygous genotypes. In the end, separate values for each season, year and marker were obtained and compared using a pairwise comparison of group means including standard errors (SE, Tukey's honestly significant difference applied R, package AGRICOLAE, Mendiburu (2019)). Based on the amplification success rate NA_{reps}, an expected error in the consensus genotype at a specific locus was quantified as NA_{reps}^{n}, whereby n was the number of total replicates.

To quantify allelic dropout at one allele, the rate of false-homozygotes was estimated as the relative number of false-homozygote replicates in consensus heterozygote genotypes. Hereby only samples for which a consensus multilocus genotype was scored were considered. Each consensus genotype was given a value, based on whether and how many false-homozygote replicates it contained. This score was either 0.33, if the genotype contained one false-homozygote of three positive PCRs, or zero, if it did not contain any false-homozygote replicates. For samples containing more than two false-homozygotes in the replicates, no consensus genotype was determined, and thus these samples were not considered. These calculations were done separately for each year, whereby season, age of feces and marker were considered as variables. Differences were assessed based on a comparison of means in R using the package AGRICOLAE (Mendiburu 2019).

Identification of unique genotypes

I identified unique genotypes using the ALLELEMATCH package in R, which considers genotyping errors and missing data during the assignment of individuals (Galpern et al. 2012). First, I created an *amDataset* object from the consensus genotype table containing all samples from all seasons and years. Then, I ran the function *amUniqueProfile*, using the criterion *doPlot = True*. This function serves the purpose of finding the optimal criterion of discrepancy to identify unique individuals. The function allows the user to determine an optimal value through displaying a range of values for the parameter (*alleleMismatch*, Galpern et al. (2012)). With increasing number of mismatches, the number of multiple matches increases as well, and the number of unique genotypes decreases (Galpern et al. 2012). Based on the results of the plot, as well as the assumption that genotypes may contain errors (Taberlet et al. 1999), the analysis done to identify unique genotypes was based on the output generated for two mismatched alleles. Finally, I applied the function *amUnique* (*alleleMismatch = 2*) to the *amDataset* object, which results in a table containing unique genotype groups, mismatch values, sample numbers, and other parameters (Galpern et al. 2012). I then analyzed the table, whereby I double checked genotypes showing mismatches using two additional, previously excluded loci (Sat2 & Sat12). Checking the additional loci was done qualitatively through comparing peak patterns as phenotypes in GENEMAPPER v5.0 (Thermo Fisher Scientific) and not through allele scoring. Some examples of phenotype classified as the same individual are given in Appendix 1.5. In addition, I checked multi-match samples analogously to find their true genotype group. Multi-match samples show a match to multiple unique genotype groups (Galpern et al. 2012). Unclassified samples are samples which are not sufficiently different from other genotypes to be considered unique, while not being similar enough to any unique genotype group

(Galpern et al. 2012). I compared unclassified samples manually to all other samples for the seven loci considered, taking into account a mismatch level of two alleles. When showing matches at five or more loci, I checked them using the additional loci under the procedure described above. In general, I considered samples to belong to the same unique group of multilocus genotypes if they did not show more than two mismatched alleles including the additional loci.

Further, I double-checked female genotypes assigned to male groups using the same procedure for the additional loci. Females showing an exact match to other male genotypes were classified as false females. Considering that some females were classified as false females, samples scored as females but not showing any match to other samples were considered as unknown sex, because their sex could not be confirmed based on multiple samples. Samples that were scored as unique genotypes in the whole dataset were compared again to all other genotypes to make sure they were truly unique. I only considered them unique, if they were different than all other samples at more than two loci, including the additional loci (Sat2, Sat12). First, a group ID equal to the first sample number was given to each group of unique genotypes. Second, unique genotype groups were sorted based on their group ID and numbered from one to the number of unique individuals. Thus, the first sample found for an individual determines their individual ID and in turn gives information about the sampling session in which the individual was first observed. To see whether more samples were collected for individuals with first observations in earlier years (lower individual IDs), I constructed a linear model in R and plotted the results including a 95% Confidence Interval.

Identification of unique genotypes as well as quantifications of error rates were done using the whole dataset, containing additional samples from Buffalora (GR, CH), Lenzerheide (GR, CH) and around the study area. After identifying the unique genotypes, I cropped the dataset in ARCGIS v.10.5.1. (Esri ARCMAP v10.5.1, USA) to obtain a dataset containing only samples collected in the study area. All subsequent analyses were done using this cropped data set. Thus, Individual IDs have a larger range than the number of individuals observed in the study area.

Observation of individuals

To achieve an overview of the time the individuals spent in the study area and the minimum lifespan, I looked at the number of samples found for each individual (recapture rate) in each season as well as the timespan across which individuals were observed. Further, I assessed whether samples are found in all sessions in which the hare is presumed as present or whether gaps occur in the observance of individuals. In addition, I estimated differences in the number of sampling sessions individuals each sex was genotyped successfully using a comparison of means (AGRICOLAE, Mendiburu, 2019).

Comparison of sampling methods

Sampling was done opportunistically and systematically (Rehnus & Bollmann 2016). To compare the efficiency of both sampling methods over an extended period, I divided the observations into opportunistic and systematic samples based on the coordinates of their origin. With this data, I compared both methods on a sample and individual level. For both sampling schemes I calculated the number of samples collected, the number of unique individuals, sex ratio and recaptures. I calculated the male-female sex-ratio as the number of males divided by the number of females. All calculations were done for spring and fall separately and were calculated as the number of individuals resp. observations per session, averaged across all years. I calculated the mean across years for each session to get a quantification of the results of the efforts of one sampling session as an average.

Population genetic statistics

I calculated probabilities of identity under the assumption that individuals are unrelated (P_{ID}) as well as in consideration that there may be siblings in the data (P_{IDsib}), the number of alleles per locus and overall, and tested for deviation from Hardy-Weinberg equilibrium (Hardy 1908, Weinberg 1908) and the presence of null alleles with CERVUS v3.0.7 (Kalinowski et al. 2007). To calculate the estimated frequency of null alleles, CERVUS uses an iterative algorithm, which is based on observed and expected frequencies of the various genotypes (Summers & Amos 1997). If no null alleles are present, the value given by CERVUS will be around zero and can be slightly negative, which indicates an excess of heterozygotes, or slightly positive. Large positive values indicate an excess of homozygotes, which however does not necessarily mean that null alleles are present (Kalinowski et al. 2007). Consequently, I included loci showing high values for estimates of null allele frequencies in the analysis, because (i) Cervus does not give true proof of null alleles (Kalinowski et al. 2007), (ii) high frequency of homozygote genotypes may appear by chance if only a few alleles are very common and (iii) the presence of null alleles can be integrated as error in pedigree analysis and (iv) they do not contribute much to individual identification, but the occurrence of a rare allele may be valuable information.

Capture-Mark-Recapture analysis

Capture-Mark-Recapture (CMR) methods seek to estimate animal abundance via capture of individuals (e.g. collection of genetic material) and marking (e.g. genetic identification) for future identifications. For population abundance estimation in a CMR framework, parameters are estimated through the proportion of marked and unmarked individuals in multiple sampling occasions. The simplest form thereof is the Lincoln-Peterson estimator (Pollock 2000). Some of the assumptions thereby are that the population is closed between sampling occasions and individuals have equal capture and recapture probabilities. Models have been developed, in which these assumptions can be violated. For example, the Jolly-Seber estimator (Jolly 1965, Seber 1965) provides estimates of apparent survival and the number of births, if sampling occurs during more than three intervals and in open populations.

Pollock (1982) introduced the robust design, which is a combination of open and closed models. The robust design allows the estimation of parameters under open population conditions while still estimating capture rates for each sampling session separately under closed conditions. It distinguishes between primary and secondary sampling sessions (Kendall et al. 1997). Primary sessions are separated by enough time for population change to happen, and each primary sampling session consists of one or more secondary sampling sessions. Each secondary sampling session is separated by a short timespan during which the population can be assumed closed. This allows the estimation of parameters associated with population change while accounting for capture effects which could bias the estimation of population abundance. The robust design further allows the estimation of additional parameters, such as temporary emigration and immigration (γ' and γ'', Kendall et al. (1997)), population growth and recruitment between primary sampling sessions (Pradel 1996).

Here, I applied the robust design model to estimate demographic parameters using the program MARK v9.0 (White & Burnham 1999). To do this, I set up models and ranked them using Akaike's Information Criterion with correction for small sample sizes (AICc, Burnham & Anderson (2002)). The models in MARK estimate parameters based on the information of observations of the individuals during primary and secondary sampling sessions. Parameters estimated here are φ as approximation for survival, capture (c) and recapture (p) rate and estimates for temporary emigration (γ') and immigration (γ'') and include confidence intervals.

The different models were set up to test for:

1) variation of φ by gender and time
2) variation of p and c by gender and time
3) values for γ′ and γ″

After choosing the most informative model, the parameters were calculated (i) across the most informative models as means including standard deviation (SD) between models and (ii) as estimated by the most informative model including standard error calculated by the model. In addition to estimating mean survival between primary sampling sessions (seasonal survival), apparent survival rates from spring to spring (annual survival) were calculated, as product of both seasonal survival rates. For example, apparent annual survival from spring 2014 to spring 2015 was estimated as the product of the apparent survival from spring to fall 2014 and from fall 2014 to spring 2015. The rates calculated were visualized in R using the package GGPLOT2 (Wickham 2016).

Pedigree analysis

Pedigree analysis can be used to assess the effects of contemporary landscapes on current dispersal and gene flow patterns (Holderegger & Wagner 2008, Kormann et al. 2012). Here, the goal was to determine, whether parent-offspring relationships could be detected and to assess the degree of relatedness among individuals observed in the study area. Recent developments in pedigree reconstruction methods make it possible not only to derive parent-offspring relationships, but to assess also sibship cohorts in the absence of parents in the data (Wang 2004, Wang & Santure 2009). Consequently, siblings in the area could theoretically be detected, even if the parents have passed away or dispersed.

Pedigree analysis was done with COLONY v2.0.6.5 (Jones & Wang 2010, Wang 2004, Wang & Santure 2009). COLONY implements new error models for markers with high amounts of false alleles and allelic dropout and is still able to achieve high likelihoods for full-siblings and parent-full-sibling cohorts (Wang 2019). Further, COLONY allows for polygamy for both females and males in addition to deviations from Hardy-Weinberg equilibrium. Polygamy as implemented by COLONY describes the fact that one male could be the father of multiple offspring with multiple females as mothers in one dataset, which does not have to be only one generation. COLONY uses data inputs for candidate offspring individuals (CO), candidate mothers (CM) and candidate fathers (CF) to assign sibship and parentage concurrently (Wang & Santure 2009). COLONY gives information on clusters consisting of related individuals, and their probability (P_C). I chose to do different runs accounting for different values for the implemented parameters and different types of data input. First, all 96 unique genotypes (see Results) detected in the study area were used to infer parentage relationships without any separate exclusions of maternal or paternal relationships (Run 1). For the second run, individuals detected for the first time more than two sessions after other individuals, were excluded as their candidate parent (CP) (Table 1). As data input for COs and CPs all individuals were used; exclusions are given as separate data input.

For Run 3 – 7, a reduced data set was used for CPs and COs (Table 2). This was done to check for differences among the clusters in the outputs of different runs and to see whether probabilities change if the amount of data input changes. Through multiple runs, one individual was able to be offspring in one output and a parent in the next. Considering the run including the whole dataset, one individual can only be either offspring or parent.

In addition to changing the data input and including exclusion of some CMs and CFs, COLONY allows to change other parameters. One parameter I changed with different runs is the probability of the dataset to include fathers (P_F) and mothers (P_M). As the study area covers a large altitudinal

gradient and a large area, and most mountain hares show high site fidelity (Dahl & Willebrand 2005), I made the assumption that most individuals stay in the area. Considering this assumption, two different analyses were run with different proportions of sampled parents. A large probability was chosen under the assumption that most parents were detected (0.9) and a second run was done with the suggested value (0.5) by Jones & Wang (2010). All seven runs described were run once with the higher and once with the lower value and P_F was always set equal to P_M ($P_M = P_F = P_P$).

Further, two different error rates were considered. First, the error rate calculated (expected error, accounting for differences across loci, see below) based on missing data in the replicates was considered. Second, I extrapolated the calculated error rate, in a way that the minimum error rate was 0.2, keeping the relative differences across loci as before (Table 3). The goal of the extrapolation was to consider a higher error rate of the loci with the goal to (i) quantify the impact of error rates and (ii) account for the possibility that error rates could be underestimated. In summary, the following four conditions were applied to all seven runs:

a. *proportion of sampled parents*:
   1) $P_F = P_M = 0.5$
   2) $P_F = P_M = 0.9$
b. *error rates*:
   3) $E \geq 2\%$
   4) E = expected error (rate calculated)

Summary of Run (1-7) described above:

1) No exclusions (all samples are COs, all females/males/unknown are CPs)
2) COs/CPs = all individuals; exclusions as described in Table 1
3) Reduced data input runs; no separate exclusions (Table 2)

For the analysis of the pedigree results, I assumed clusters consistent across runs with $P_C \geq 0.8$ (significant clusters) in at least one run to be true.

Table 1: Exclusions applied in Run 2 based on the whole data set. For each CO, CPs for the season "Excluded" and after were excluded as potential parents.

				Sampling session				
2014S	**2014F**	**2015S**	**2015F**	**2016S**	**2017S**	**2017F**	**2018S**	**2018F**
CO	√	√	Excluded					
√	CO	√	√	Excluded				
√	√	CO	√	√	Excluded			
√	√	√	CO	√	√	Excluded		
√	√	√	√	CO	√	√	Excluded	

Table 2: Run number and data input for runs 3 – 7 in COLONY: If the data input for COs consists of all offspring observed in fall 2014 and later (2014F+), then CPs consisted only of individuals detected for the first time in spring 2014 (2014S).

Run	**1st observation of**	
	CO	**CP**
3	2014F +	2014S
4	2015F +	2014F – 2015S
5	2016F +	2014F – 2016S
6	2017F +	2014F – 2017S
7	2018F (+)	2014F – 2018S

Table 3: Extrapolated error rates (≥ 0.02)

Lsa1	**Lsa3**	**Sat5**	**Sat8**	**Sol30**	**Sol33**	**Sol8**
0.0605	0.1036	0.072	0.02	0.058	0.083	0.037

Results

Genotyping

Over the study years, 1588 samples were collected in and around the study area in the Swiss National Park. Of these samples 331 (20.7%) had to be removed because they showed more than one missing locus in the determined consensus genotype. Some samples (N = 44) were collected in other areas around the study area.

Error statistics

The amount of missing data in these samples varied significantly between sessions, whereby samples collected in fall showed a higher amount of missing data than samples collected in spring ($p = 0.001$, Figure 2). In addition, the rate of false homozygotes differed between spring and fall sampling sessions ($p = 0.001$). However, the proportion of false homozygotes was larger in spring than in fall (Figure 4). Further, loci containing a low amount of missing data did not necessarily show a low false homozygote rate (Figures 1 & 3). For example, locus Lsa3 contained many missing replicates (0.1433 ± 0.0357, mean ± SE) but did not contain many false homozygotes (0.0033 ± 0.0009). Overall, the proportion of false homozygotes varied significantly between loci ($p < 2.2 \cdot 10^{-16}$) and genotyping success (missing replicates) did not ($p = 0.8896$). Locus Sat8 showed both a low number of missing replicates (0.0828 ± 0.03) and a low proportion of false homozygotes (0.0013 ± 0.0006) across years and the season of sampling sessions. The locus containing the highest number of false homozygote replicates was locus Sol33 (0.0244 ± 0.0026), which also contained a relatively high number of missing replicates (0.1334 ± 0.0357).

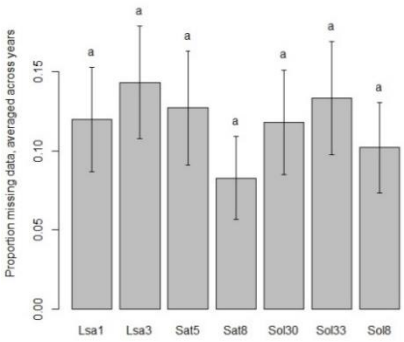

Figure 1: Means and SE for the amount of missing data per locus.

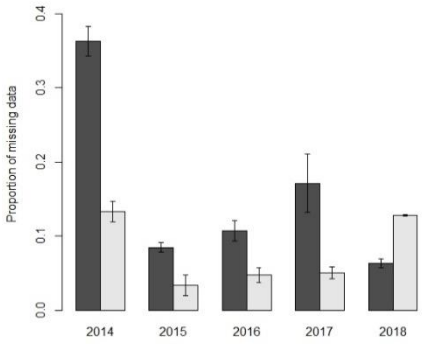

Figure 2: Means and SE for the amount of missing data detected in each year and season (fall = dark-grey & spring = light-grey).

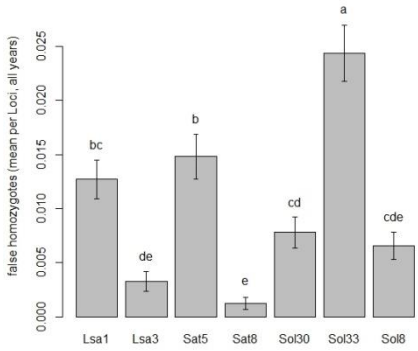

Figure 3: Means and SE for the number of false homozygotes per loci.

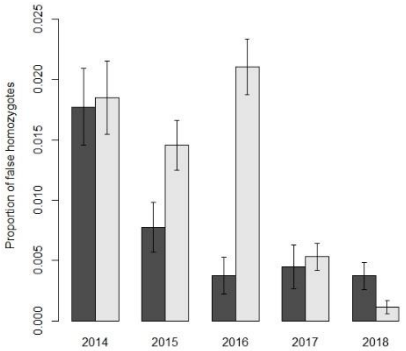

Figure 4: Means and SE for the number of false homozygotes detected in each year and season (fall = dark-grey, spring = light-grey).

Table 4: Amount of missing replicate genotypes (NA_{reps}) in the whole data set, expected error in the consensus genotypes (NA_{resp}^{3}) and proportion of false homozygote replicates in the consensus multilocus genotypes for each locus.

Locus	NA_{reps}	Expected Error	False Homozygotes
Lsa1	0.12 ± 0.03	0.0017	0.0127 ± 0.0018
Lsa3	0.14 ± 0.04	0.0029	0.0033 ± 0.0009
Sat5	0.13 ± 0.04	0.0021	0.0148 ± 0.0021
Sat8	0.08 ± 0.03	0.0006	0.0013 ± 0.0006
Sol30	0.12 ± 0.03	0.0016	0.0078 ± 0.0014
Sol33	0.13 ± 0.04	0.0024	0.0244 ± 0.0026
Sol8	0.10 ± 0.03	0.0011	0.0066 ± 0.0013

Identification of unique genotypes

Running the *amUniqueProfile* function in ALLELEMATCH revealed two mismatches to be the optimal value for identifying unique genotypes, as a result of the minimum number of samples with multiple matches being at this point. In addition, the number of unique genotypes was estimated to be at an intermediate value (Figure 5). Based on this output, the *amUnique* function revealed 90 unique genotypes for two allele mismatches. Additionally, 37 samples were unclassified, and 153 samples showed a match to multiple groups.

After double checking the output of *amUnique* through consideration of the two additional loci Sat2 and Sat12 (see Appendix 1.4), 118 unique genotypes were identified in all samples. Of the 1268 samples resulting in 118 genotypes, 1224 samples were found in the study area, which were assigned to 96 individuals. Thereof, 17 samples were scored as single unique genotypes, i.e. for 17 individuals only one sample was found. Contrastingly, for one individual a large number of samples were found (N_{max} = 106). However, for most individuals an intermediate number of samples were collected (N_{mean} = 13.13, Figure 6). Individual IDs were assigned chronologically to genotype groups in accordance with sample collection, i.e. the first sample collected from a genotype group determined its individual ID. Thus, individuals that only appeared in later years were assigned higher individual IDs. For most individuals only detected in later years, fewer samples were found than for individuals with first observations in early sampling sessions. (p = 0.002, Figure 6).

No difference was found between the number of samples collected for females and for males, however, the mean number of samples was slightly higher for females than for males (Figure 7). However, male samples include individuals for which only one sample was found, whereas for females these samples were excluded and defined as unknown sex. Consequently, the category of unknown sex includes only observation numbers equal to one.

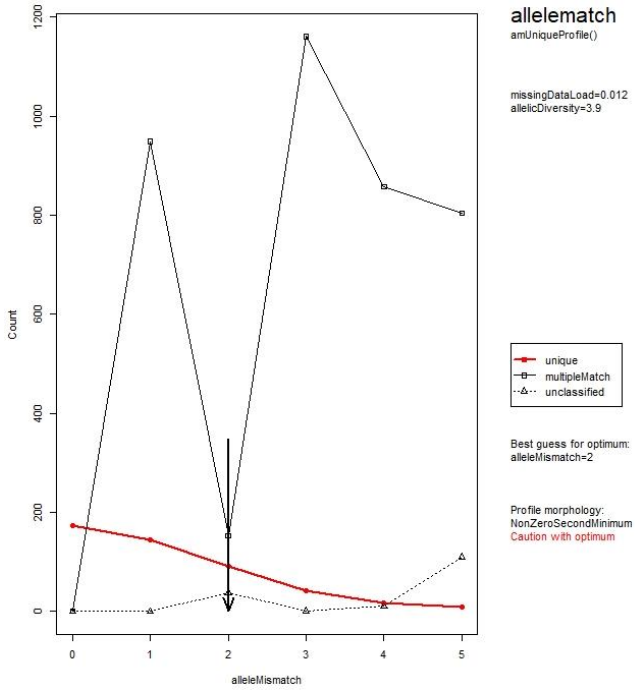

Figure 5: Resulting plot of the *amUniqueProfile* function, applied to the whole dataset (N = 1588), excluding samplings with more than two loci missing in the consensus genotype.

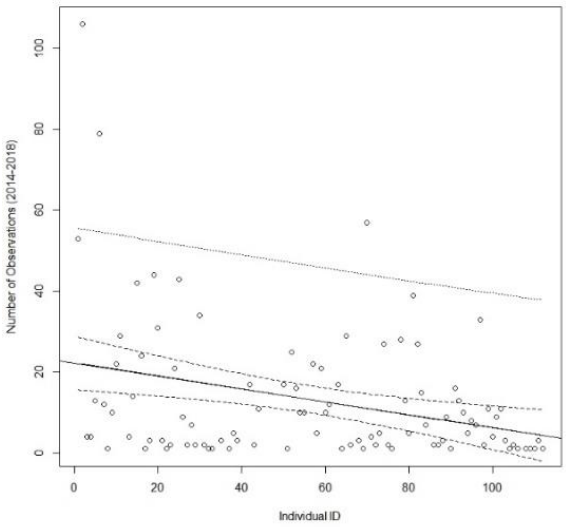

Figure 6: Linear regression model showing the correlation between the number of observations throughout all study years and the individual ID (including 95% CI, p = 0.002).

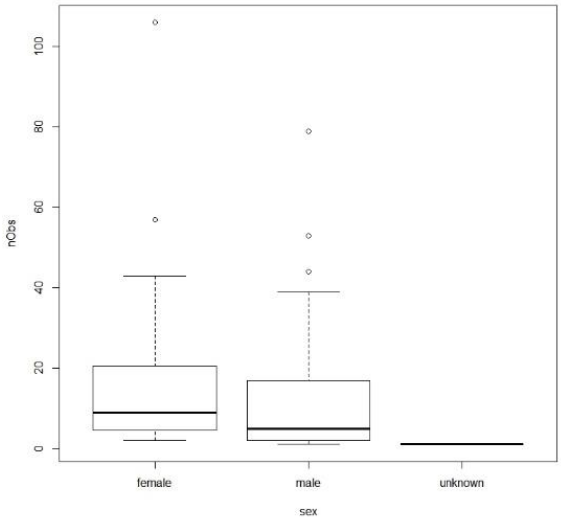

Figure 7: Mean number of observations (nObs) per sex. Unknown sex individuals contain only one single observation per individual.

Observations of individuals

In total, 31 females, 59 males and six individuals of unknown sex were detected in all study years. More individuals were observed each spring ($\widehat{N}$ = 25.4) than fall ($\widehat{N}$ = 21.0). In addition, individuals were observed more often (i.e. more samples were found per individual) in spring than in fall (Table 5). The number of samples found for an individual in one sampling session varies from 1 – 29. Individuals were mostly observed multiple times throughout the whole study time. However, only one individual (IndID02, female) was observed in every sampling session. Most individuals detected during multiple sessions were detected in all consecutive sessions, without many missing observations from sessions in between (Figure 8). As all female individuals which were only detected by one sample were categorized as unknown sex, all individuals of unknown

sex were only observed once during one sampling session. Apart from that, no obvious patterns between females and males were distinguished.

The number of individuals detected in each session varies between 17 (fall 2017) and 28 individuals (spring 2017), with a mean of 23.8 individuals per session. Thereby, more males than females were detected in most sampling sessions, whereas in spring the number of males was always larger than the number of females (Table 5). The calculated mean sex ratio (male:female) was consequently larger in spring (1.63) than in fall (1.11, Table 6).

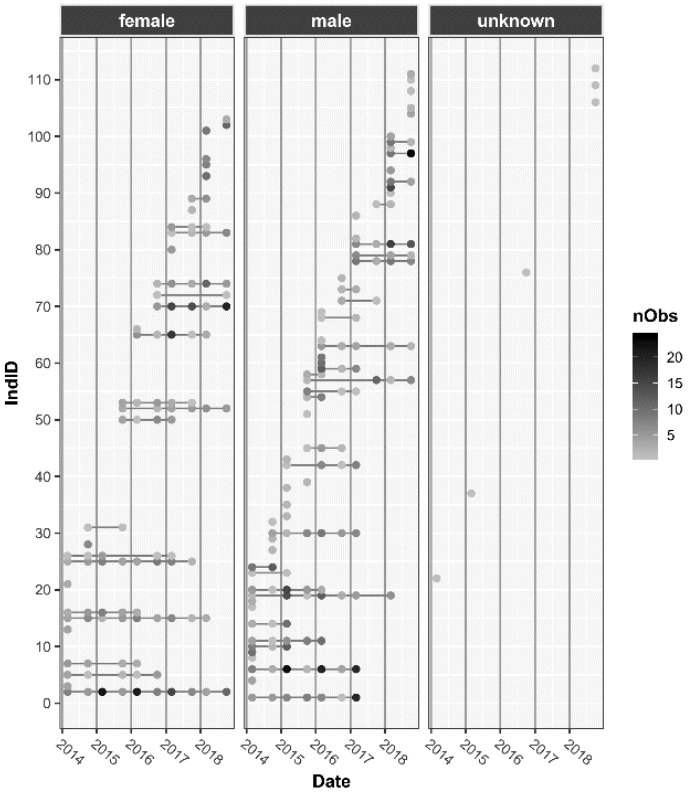

Figure 8: Individual observations in the years 2014 – 2018. Dots represent single observations of individuals; repeated observations of the same individual are connected by a line.

Table 5: Number of individuals detected in each sampling session in the study area

	2014		2015		2016		2017		2018	
	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall
N_F	10	9	5	10	10	12	13	12	13	8
N_M	14	11	14	13	17	12	15	5	15	13
N_{NA}	1	0	1	0	0	1	0	0	0	4
N	25	20	20	23	27	25	28	17	28	25

Sampling methods comparison

Of all genotyped samples found in the study area, more samples were collected opportunistically (N = 992) than systematically (N = 232). For both methods, the average number of samples found in spring (34 resp. 122.4) was higher than in fall (12.4 resp. 73.4, Table 6). Additionally, more individuals were detected by opportunistic sampling (N = 85, 91%) than by systematic sampling (67, 52%) and the number of recaptures was larger for systematic sampling (2.21) than for opportunistic sampling (6.30, Table 6). Nevertheless, most individuals were detected by both methods, and merely some individuals were only detected by either systematic or opportunistic

sampling. More precisely, opportunistic sampling detected 30 individuals which were not detected through systematic sampling and systematic sampling found nine individuals which were not detected through opportunistic sampling.

.

Table 6: Results of the comparison of sampling methods: The results for separate seasons were calculated as the mean of each season across all years.

spring	Sampling		
Parameter	Systematic	Opportunistic	Combination
Number of samples	34	122.4	788
Number of unique individuals	14.2	23.6	25.4
Sex ratio (male:female)	1.94	1.08	1.63
Recaptures per individual	2.30	5.06	10.65

fall	Sampling		
Parameter	Systematic	Opportunistic	Combination
Number of samples	12.4	73.4	436
Number of unique individuals	9.2	18.8	21
Sex ratio (male:female)	1.59	1.48	1.11
Recaptures per individual	1.34	3.88	6.92

total	Sampling		
Parameter	Systematic	Opportunistic	Combination
Number of samples	232	992	1224
Number of unique individuals	67	85	96
Sex ratio (male:female)	1.56	1.24	1.84
Recaptures per individual	2.21	6.30	7.03

Table 7: The number of females (F) and males (M) detected by systematic (S) and opportunistic (O) sampling and in total (N_F, N_M); given in absolute (N) and proportional numbers (D) to the total detected by both methods. The total gives the absolute number of males (NM) and females (NF) detected across all study years and the proportion thereof detected by each sampling method.

	2014		**2015**		**2016**		**2017**		**2018**		**Total**
	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	
N_{FO}	10	7	5	10	10	11	12	10	14	7	
N_{MO}	12	9	12	11	15	10	15	4	14	13	
N_{FS}	4	5	5	1	3	6	10	5	6	7	
N_{MS}	7	4	9	5	12	6	5	3	10	4	
N_F	10	9	5	10	10	12	13	12	14	8	31
N_M	14	11	14	13	17	12	15	5	15	13	59
D_{FO}	1,00	0,78	1,00	1,00	1,00	0,92	0,92	0,83	1,00	0,88	0,93
D_{MO}	0,86	0,82	0,86	0,85	0,88	0,83	1,00	0,80	0,93	1,00	0,88
D_{FS}	0,40	0,56	1,00	0,10	0,30	0,50	0,77	0,42	0,46	0,88	0,54
D_{MS}	0,50	0,36	0,64	0,38	0,71	0,50	0,33	0,60	0,67	0,31	0,50
D_O	0,93	0,80	0,93	0,92	0,94	0,88	0,96	0,82	0,97	0,94	0,91
D_S	0,45	0,46	0,82	0,24	0,50	0,50	0,55	0,51	0,56	0,59	0,52

Population genetic statistics

In contrast to the missing data detected in the replicated samples, the missing data in the consensus multilocus genotypes was unevenly distributed among loci. Loci Lsa3, Sat5, Sol33 and Sol8 contained missing genotypes, whereas Sat8 and Sol30 did not. Across all samples, an average number of 4.4 alleles per locus were detected. The highest allele count was found for locus Sat5, the lowest number of alleles was found at locus Sat8 (Table 8).

The results of the calculations done to estimate null allele frequencies show Lsa3 and Sat5 to contain an excess of homozygotes resp. potential null alleles (Table 8). Lsa3 was estimated to contain a null allele frequency of around 46% and Sat5 a frequency of approximately 11%. At locus Lsa3, allele 210 was by far the most common ($P_{210} = 0.78$) and most genotypes containing this allele were homozygous (~83%). For both loci the estimated null allele frequency indicates the presence of null alleles resp. an excess of homozygotes. However, these loci were still included in the analysis, as no true proof of null alleles can be given and estimates of null allele frequencies in CERVUS are based on homozygote frequencies (Kalinowski et al. 2007). An excess of homozygotes may also appear by chance if one allele is a lot more common than other alleles. Additionally, even if null alleles were present, individual identification and pedigree reconstructions may still be possible if proper error models are integrated (Dakin & Avise 2004, Wagner et al. 2006, Wang 2019).

High values of P_{IDsib} as well as P_{ID} per locus were estimated for loci Sat8 ($P_{IDsib,Sat8} = 0.87$) and Lsa3 ($P_{IDsib,Lsa3} = 0.67$). For these loci a low number of alleles ($A_{Sat8} = 2$) or a low frequency of observed heterozygosity ($H_{O,Lsa3} = 0.15$) was found. Woods et al. (1999) applied a maximum across-loci value of P_{IDsib} of 0.05 for individual identification, which is larger than the P_{IDsib} calculated for this dataset ($P_{IDsib,across} = 0.036$, Table 8).

Table 8: Number of alleles (A), number of individuals genotyped (N), observed and expected heterozygosity values (H_O, H_E), probabilities of identity (P_{ID}, P_{IDsib}) and estimates of null allele frequencies ($E(F_{NULL})$) given for each locus and across loci across all study years.

Locus	A	N	H_O	H_E	P_{ID}	P_{IDsib}	$E(F_{NULL})$
Lsa1	3	96	0.677	0.611	0.231	0.504	- 0.0567
Lsa3	5	93	0.14	0.371	0.416	0.669	0.4645
Sat5	7	92	0.391	0.509	0.274	0.565	0.1164
Sat8	2	96	0.146	0.136	0.757	0.872	-0.0295
Sol30	6	96	0.396	0.426	0.365	0.629	0.0375
Sol33	3	91	0.582	0.585	0.264	0.525	- 0.0010
Sol8	5	95	0.389	0.392	0.43	0.662	- 0.0006
Across	4.4	96	0.389	0.433	0.0008	0.036	

Capture-Mark-Recapture (CMR) calculations

Model comparison

Models in MARK (White & Burnham 1999) were setup to estimate variation of the parameters in regard to time and sex. Of the totally run models (22), 16 models were unsupported, with model weights of 0 and ΔAIC values of > 20. The most likely model had more than half of the weight of the total model set ($w_1 = 0.52$), and the six most likely models accounted for almost all the weight of the whole set ($w_1 – w_5 = 0.99$, Table 9)

The model with the best fit (Model 1) described apparent survival with variation by sex and time, equal capture and recapture probabilities, and values for temporary emigration and immigration varying only by sex. The second-best fit was achieved by the model accounting for unequal capture and recapture probabilities with variation by sex. Thus, the difference between the most likely and the second most likely model were equal or unequal capture and recapture probabilities (Table 9).

Equal capture and recapture probabilities were included in all models but Model 2. One of six supported models (Model 4) included fixed parameter values for temporary immigration and emigration. All other supported models included estimates for temporary immigration and emigration with variation by either sex (Model 1 & 2) or time (Model 3 & 5) or both (Model 6).

Estimates of apparent survival either included variation by time and sex (Model 1, 2 & 4), only sex (Model 3 & 5), or constant values (Table 9).

Table 9: Model comparison of the models obtained in MARK for models 1 to 6, including values for AICc, ΔAIC, AICc Weight, model likelihood and the number of parameters estimated. Estimated parameters are apparent survival rate (φ), temporary immigration (γ″) and emigration (γ′) and capture (c) and recapture (p) probabilities

Model No.	Model	AICc	ΔAIC	AICc Weight	Model Likelihood	No. Par.
1)	φ(t,sex), γ″(sex), γ′(sex), p(sex) = c(sex)	897.6111	0.0000	0.51561	1.0000	25
2)	φ(t,sex), γ″(sex), γ′(sex), p(sex), c(sex)	898.6117	1.0006	0.31264	0.6064	27
3)	φ(sex), γ″(t), γ′(t), p(sex) = c(sex)	900.6225	3.0114	0.11439	0.2219	22
4)	φ(t,sex), γ″=0, γ′=1, p(sex) = c(sex)	902.7025	5.0916	0.04043	0.0784	21
5)	φ(.), γ″(t), γ′(t), p(sex) = c(sex)	904.4517	6.8406	0.01686	0.0327	21
6)	φ(sex), γ″(t, sex), γ′(t, sex), p(sex) = c(sex)	915.6114	18.0003	0.00006	0.0001	38

Parameter estimates

Mean apparent seasonal survival estimates across Models 1 – 4 and standard deviations (SD), showed that the models estimated similar values (Table 10). Across models, apparent seasonal survival was estimated to be higher from spring to fall (0.92 ± 0.02, estimate ± SD) than from fall to spring (0.67 ± 0.03). The apparent annual survival calculated based on models 1 – 4, was estimated to be larger for females than males in all study years (Table 10).

The model with the best fit (Model 1) estimated average apparent seasonal survival throughout the whole study time to be approximately the same for females (0.79 ± 0.12, estimate ± SE) and males (0.77 ± 0.12). However, the estimates for males showed higher fluctuations than for females across the study years (Figure 9). Additionally, the probability to reappear in the study area at time t + 1, after not having been observed at time t (γ″) was larger for males (0.17 ± 0.07) than for females (0.09 ± 0.05). Thus, the probability ~~of a male~~ to be unobserved at time t and rest unobserved at time t + 1 was smaller for males (0.83) than for females (0.91). The model thus estimated that males are out of the study more frequently and return, whereas females showed higher site-fidelity. Females were more prone to remain off the study at consecutive sampling sessions ($P_{females}$ = 0.45 ± 0.28) than males (P_{males} = 0.32 ± 0.29) after not having been observed in the previous session. However, the standard error of these parameters, as estimated by the model, was relatively high.

The probability of becoming observable at time t + 1 when not having been observed at time t was larger for males (P_{males} = 0.68) than for females ($P_{females}$ = 0.55). Therefore, a male was more likely to reappear at time t+1, after not having been observed at time t, than a female.

Capture probabilities for males (0.72 ± 0.05) were estimated to be marginally lower than for females (0.77 ± 0.04). Models 1 – 4 estimated similar values, and model means were 0.74 ± 0.02 (estimate ± SD) for females and 0.69 ± 0.02 for males.

Table 10: Estimates of apparent survival as means across models 1 – 4 with the standard error (variation) between models. All uneven numbers describe apparent survival from spring to fall and all even numbers describe survival from fall to spring. E.g., parameter S1 describes the apparent survival from sampling season 1 to sampling season 2 (spring 2014 spring to fall 2014). Additionally, apparent annual survival estimates across models are given, whereby $S_{i\text{-}y}$ gives the survival from year i to year y.

Females		**Males**	
Parameter	**Estimate ± SE**	**Parameter**	**Estimate ± SE**
S1	0.92 ± 0.04	**S1**	0.66 ± 0.05
S2	1.00 ± 0.00	**S2**	0.72 ± 0.03
S3	0.88 ± 0.01	**S3**	0.45 ± 0.01
S4	0.95 ± 0.00	**S4**	0.99 ± 0.01
S5	0.72 ± 0.00	**S5**	0.64 ± 0.02
S6	0.91 ± 0.01	**S6**	0.98 ± 0.02
S7	0.84 ± 0.01	**S7**	0.37 ± 0.00
S8	0.87 ± 0.01	**S8**	0.91 ± 0.01
S9	0.46 ± 0.01	**S9**	0.79 ± 0.05
$S_{14\text{-}15}$	0.92 ± 0.04	**$S_{14\text{-}15}$**	0.47 ± 0.05
$S_{15\text{-}16}$	0.84 ± 0.01	**$S_{15\text{-}16}$**	0.44 ± 0.01
$S_{16\text{-}17}$	0.66 ± 0.01	**$S_{16\text{-}17}$**	0.62 ± 0.02
$S_{17\text{-}18}$	0.73 ± 0.01	**$S_{17\text{-}18}$**	0.33 ± 0.01

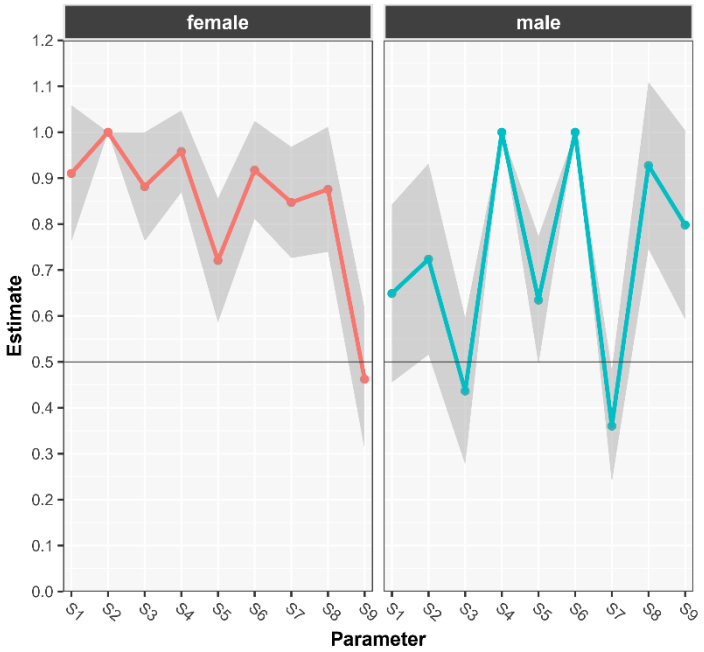

Figure 9: Apparent survival as estimated by Model 1 including standard error estimated by the model.

Pedigree analysis

Comparison of runs

Only a few significant clusters were found to be consistent across the different runs. The number of significant clusters (NC_{08}) using the whole dataset with or without exclusions differed depending on the error rates supplied and the P_P chosen for the run. Across these runs, the highest number of significant clusters were found for the run using exclusions, low error rates and a P_P of 0.5 (Table 11).

The run using exclusions, a low P_P and low error rates led to the identification of nine clusters, of which three were significant. High error rates, low P_P and no exclusions led to the identification of two significant clusters from a total of seven clusters (Table 11). The combination of high error rates and a high parent probability did not lead to any significant clusters in any run. Significant clusters using high error rates were only obtained when a low P_P was chosen. Consequently, the impact of P_P on the number of total and significant clusters was larger if high error rates were used (Table 11). The number of significant clusters in relation to the number of total clusters was larger for low than for high error rates (Figure 10).

For the runs using only a reduced amount of data input, more significant clusters were found for low error rates than for high error rates. Further, analogously to the runs with all data, when high error rates were applied more clusters were found for a low value of P_P compared to a high value of P_P. High error rates and high values for P_P did not lead to any significant clusters and only three total clusters. However, when low error rates were used, higher values for P_P lead to the identification of more clusters than lower values. For the first run, three significant clusters from a total of six clusters were identified using a high P_P and low error rate estimates.

Table 11: Overview of the total number of clusters (NC) and the number of significant clusters (NC_{08}) obtained with different parameter inputs: Exclusions, extrapolated high error rates or low error rates, and different probabilities of parents to be in the data set (P_P)

NC	NC_{08}	Exclusions	Error Estimates	P_P
9	3	yes	low	0.5
7	2	none	high	0.5
6	2	none	low	0.5
4	2	yes	low	0.9
3	2	none	low	0.9
5	0	yes	high	0.5
2	0	none	high	0.9
2	0	yes	high	0.9

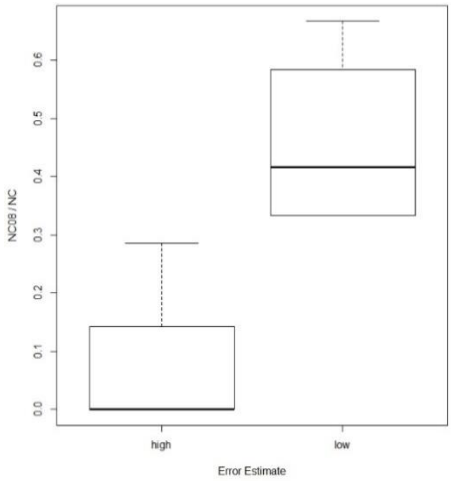

Figure 10: Relative proportion of significant clusters found per total number of clusters with high and low error rates.

Identification of significant clusters

The number of clusters as well as most of the individuals in the respective clusters differed between runs. No individuals could be determined as parents with high probabilities in any runs. However, a few clusters showed consistencies among runs and some sibling-relationships could be reconstructed with acceptable probabilities. The most consistent cluster signified that individual 60 was different from the other individuals, as individual 60 formed a separate cluster in seven runs based on low error rates and different parent probabilities. Individual 60 formed a full-sib cluster with individuals 61 and 69 in one run considering high error rates and a high value for P_P. Individuals 61 and 69 formed a full-sib cluster in six runs, of which in five runs individual 60 was not assigned to the same cluster. Individuals 44, 66, 101, 105, and 112 were estimated to form a full-sib cluster with probabilities varying from 0,999 to 0,87 between runs. Runs with high error rates did generally not lead to the estimation of different clusters than runs with low error rates. For example, individuals 101, 105 and 112 were estimated to be siblings ($P_C = 0.8041$) based on high error rates with reduced data, as well as based on low error rates using the whole data set ($P_C = 0.999$) and the whole data set with exclusions ($P_C = 0.871$).

Methods

Tissue Samples

For this analysis, 90 samples from different locations across Europe were considered. Tissue samples were collected by hunters, game keepers, taxidermists, and other researchers and frozen at -20°C. Samples included different types of tissue, such as parts of the ear, paws, muscle and bones. All samples originate from either mountain hares (N_{Lt}=51), European hares (N_{Le}=36), known hybrids (N_{Hb}=2) or unknown species identity (N_{NA}=1). Samples were obtained from five different countries and with uneven sample distributions between countries (Table 12). All samples from mountain hares were collected in alpine regions and two samples from European hare originate from the eastern part of Germany (Figure 11).

Table 12: Number of samples from each country and species, whereby NA stands for unknown species, Hb for hybrids, Le represents European and Lt mountain hares. Countries are given in their official two-letter codes.

	NA	Hb	Le	Lt	Total
CH		2	19	24	45
DE	1		2	3	6
AT			15	7	22
FR				5	5
IT				12	12
Total	1	2	36	51	90

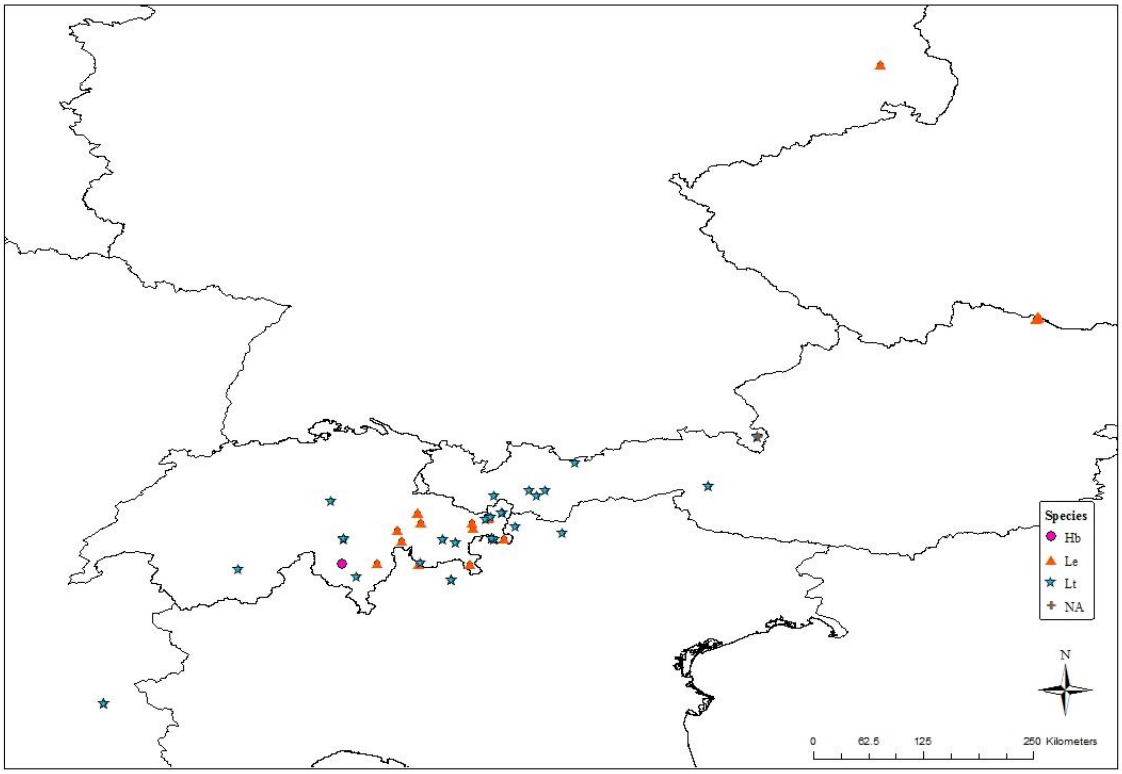

Figure 11: Sampling locations of the tissue samples for the two study species. Hybrid samples (Hb) are marked with a pink dot, European hares (Le) are displayed as orange triangles, mountain hares (Lt) are shown as blue stars, and samples from unknown species (NA) are marked with a brown cross.

DNA extraction and amplification of tissue samples

DNA extraction was done using the DNeasy Blood & Tissue Kit (Qiagen) following the protocol for purification of total DNA from animal tissues (Spin-Column Protocol, Qiagen).

Pieces 25mg consisting of muscle, tendon, bone or some hair were cut off from the samples and stored in a 2mL tube at -20 degrees until further analysis. A mastermix containing 180µL ATL Buffer and 20µL proteinase K for each sample was prepared and vortexed, then 200µL of the mix was added to each sample. Samples were then incubated in an oven at 56°C after being shaken at 700rpm (EPPENDORF THERMOBLOCK) for one hour. After a short spin, the lysate, including undissolved pieces, was transferred onto the DNeasy Mini spin column placed in a 2ml collection tube (Qiagen), mixed by centrifuging at 2112G for 1min and subsequently the column as well as residual tissue pieces were discarded. 200µl of the lysate was transferred into a 2ml tube and, if necessary, the volume was increased with additional ATL/ProtK mix to ensure a sample volume of 200µl. 4µl RNase A were added to the lysate, and after vortexing the samples were incubated at room temperature for 5min. After 15s of vortexing, 200µl of AL was added, and the samples were again vortexed thoroughly. A mastermix containing 200µl AL and 200µL ethanol (100%) per sample was prepared and vortexed. 400µl of the mix was added to each sample, immediately thereafter the samples were vortexed for 15s to ensure a homogeneous solution. The samples were then centrifuged at 660G for 20s. All the liquid in the tube (600µl) was then transferred on a DNeasy Mini spin column, placed in a 2ml collection tube, while making sure no liquid contaminates the edge of the column. After centrifuging at 2112G for 60s, the flow-through was disposed, the collection tube dried off and the column placed in the collection tube again. 500µl AW1 was added onto the column and the samples were centrifuged at 2112G for 60s. Flow-through was disposed, collection tube dried off and the column placed in the same collection tube. 500µl AW2 was added onto the column and samples were centrifuged for 3min at 3696G. Flow-through was disposed, collection tube dried off and the column placed on the same collection tube. Following a centrifugation at 3696G for 60s, the flow-through was disposed, and the column was placed in a new collection tube. Collection tubes and columns were opened and dried for 2-5min at 65°C in an oven to make sure all the EtOH dried off. After making sure the columns were dry, 100µl AE Buffer (heated to 65°C) was added onto the columns, the columns were incubated at room temperature for 60s and centrifuged at 2112G for 60s. The elution was repeated once with freshly added 100µL AE Buffer to increase the final DNA yield. Finally, 200µl of the eluted DNA was pipetted into a 1.5ml screw cap tube and stored at -20°C.

DNA concentration (ng/µl) was assessed using QUANTUS (QuantiFluor ONE System, Promega, E4871/E4870). First, a 2µl of a standard DNA (dsDNA ONE Standard, Lambda 400µg/ml = 400ng/µl) was added to 198µl of ONE dsDNADye, vortexed for 5s and incubated for 5min in the dark. In addition, a blank sample containing 198µl ONE dsDNADye solution and 1µl TE was mixed, vortexed for 5s and incubated for 5min. After, DNA concentration in both the blank and standard sample was measured to assess exactness of measurement. After, the samples were prepared using 2µl of sample DNA and 198µl of ONE dsDNADye solution. All samples were vortexed directly after mixing and subsequently stored for 5min in the dark before measuring DNA concentration. Samples containing not enough or too much DNA were measured with more DNA (3µl DNA and 197µl of ONE dsDNADye solution) resp. diluted with TE until measurable.

Assessing the quality of the DNA samples with NANODROP 2000 (Thermo Fisher) using 2µl of sample DNA. Before measuring sample DNA concentrations, a blank consisting of 2µl H_2O was first used to calibrate the machine and then measured to check for contaminations. Quality was based on the 260/280-nm ratio and concentration of the DNA as well as quantitively by the spectrum observed.

Additionally, all DNA quality (fragment lengths) was checked for all samples through electrophoresis using EZ-Vision Bluelight Dye (Amresco, LLC) on an 1% Agarose gel.

Based on the concentrations assessed using QUANTUS (Promega), DNA samples were diluted to 2.5ng/µl in a separate tube by adding H_2O to different amounts of DNA to achieve an approximate end-volume of 20µl diluted DNA.

The same markers used for the noninvasive genetic monitoring were applied (see Part I, page 9, "DNA extraction and amplification). Amplification was also performed in two multiplex PCRs. PCR volumes of totally 10µl contained 5µl HotStarTaq Master Mix (Qiagen), 0.2-0.4 µM of each primer pair, 1.4µl Merck H_2O and 2µl of diluted DNA (2.5ng/µl). Thermocycling consisted of an initial denaturation at 95°C for 15min, succeeded by 35 cycles of 30s at 95°C, 90s at 56°C, 60s at 72°C, and a final extension of 30min at 72°.

Tissue sample Genotyping

Fragment length analysis was conducted using ABI3730 genetic analyzer using LIZ500bp as an individual standard. Before analysis, samples were diluted with 90µl Merck H_2O. Electropherograms were analyzed using GENEMAPPER 5.0 (Applied Biosystems). PCRs not showing clear peaks in GENEMAPPER were repeated.

For samples for which multiple PCRs were achieved, consensus genotypes were created following the rules described in Appendix 1.2 for all seven loci. Sex was determined analogously to the procedure applied for the NGS samples (see Appendix 1.5).

Distinguishing between mountain hares and European hares

Population genetic statistics

After obtaining genotypes for all tissue samples, the number of alleles as well as, observed and expected heterozygosity, polymorphic information content (PIC) and null allele frequency estimates were obtained using CERVUS (Kalinowski et al. 2007) separately for all known *L. timidus* and *L. europaeus* samples and for all tissue samples under consideration of the species as populations. PIC gives a measure of informativeness of the locus and is calculated based on allele frequencies (Botstein et al. 1980, Hearne et al. 1992). I calculated PIC locus-specifically and as average across loci. Additionally, I calculated values for allelelic richniss (A_R) in R using the function *allel.rich* in the package POPGENREPORT (Gruber & Adamack 2014). A_R gives a value for allelic diversity at each locus under consideration of the sample size using a rarefaction method (El-Mousadik & Petit 1996). Finally, I compared the values obtained to check for differences between the species and visualized the results in R using the package GGPLOT2 (Wickham 2016).

Principal component analysis

To test whether European hares and mountain hares can be distinguished, I first conducted a Principal Component Analysis (PCA) using the genotype table obtained from the tissue samples. Second, I performed a PCA to test whether potential hybrids or European hares are contained in the samples of the noninvasively collected data from the National Park monitoring.

To implement the PCAs, I first created two different *genind* objects using the ADEGENET package (Jombart 2008) in R using (i) only the genotype table obtained from the tissue samples and (ii) the genotype tables of the NP dataset and the tissue samples. Thereafter, I was able to conduct a PCA explicitly for genotype datasets (Dray & Dufour 2007) and visualize the results with functions from the FACTOEXTRA package (Kassambara & Mundt 2017). I determined the number of factors retained in the PCA with the option *SCANNF = F*, which displays a Scree plot and allows the visual determination of the informativeness of factors (Dray & Dufour 2007). Scree plots display

eigenvalues associated with each component, which enables the identification of factors with large respectively small eigenvalues and the break in between (Cattell 1966).

STRUCTURE analysis

To get further confirmation for species associations, I assessed individual assignments using STRUCTURE v2.3 (Pritchard et al. 2000). I conducted two analyses in STRUCTURE, the first using only the tissue data genotypes as input to establish whether STRUCTURE determines assignments comparable to the PCA or whether differences occur between the two methods. Second, I used the NP data set as well as the tissue data set as input, to test for species associations of the individuals detected in the National Park.

For both analyses, I used the admixture model to determine the number of clusters (K). The admixture model was chosen because it was estimated to be reasonable that some individuals may have common ancestors (Porras-Hurtado et al. 2013). Additionally, the correlated allele frequency model was used, which provides greater power to detect distinct but closely related populations (Porras-Hurtado et al. 2013). The analysis was run using a burn-in period of 10'000 and 10'000 Markov Chain Monte Carlo (MCMC) reps after burn-in. Values for K were set from 1 – 5, using ten iterations per K.

The most probable value of K was assessed using STRUCTURE HARVESTER WEB v0.6.94 (Earl & Holdt 2012), which implements the Evanno method to estimate the most likely number of clusters (Evanno et al. 2005). After estimating the value for K with the highest likelihood, the results for the runs of the respective cluster were combined using CLUMPP v1.1.2 (Jakobsson & Rosenberg 2007) and plotted.

Results

Population genetic statistics

Population genetic statistics obtained with CERVUS revealed differences in the number of alleles and the degree of heterozygosity (observed and expected) between both species (N_{Lt} = 51, N_{Le} = 36; Table 13). The average of the absolute number of alleles was slightly higher for European hares (A = 6.7) than mountain hares (A = 5.9) across loci. Allelic richness (A_R) was also estimated to be higher for European hares than for mountain hares at all loci except Sol33 and consequently as mean across loci (Table 13). However, relative differences among allele counts were similar for both species across loci (Figure 12). Loci with a high value of A for one species also showed a high relative number of alleles in the other species (Table 13, Figure 12).

Across loci, the difference between expected and observed heterozygosity was similar between both species. However, observed and expected heterozygosity was larger for European hares (H_O = 0.522, H_E = 0.665) than for mountain hares (H_O = 0.359, H_E = 0.443, Table 13). In addition, average PIC across loci was higher for European hares and both probabilities of identity (P_{ID}, P_{IDsib}) were more than tenfold larger for European than for mountain hares. At some loci estimates of null allele frequencies were larger for *L. europaeus*, whereas at other loci estimates were higher for *L. timidus*. For example, concerning European hares, null alleles were estimated to be present at loci Sat5 and Sol33. For mountain hares, null alleles were estimated to be present at locus Sat5 but not at locus Sol33. Overall, the values indicated the presence of null alleles at loci Lsa3, Sat5, and Sol33 for European hares and at loci Lsa3 and Sat5 for mountain hares (Table 13).

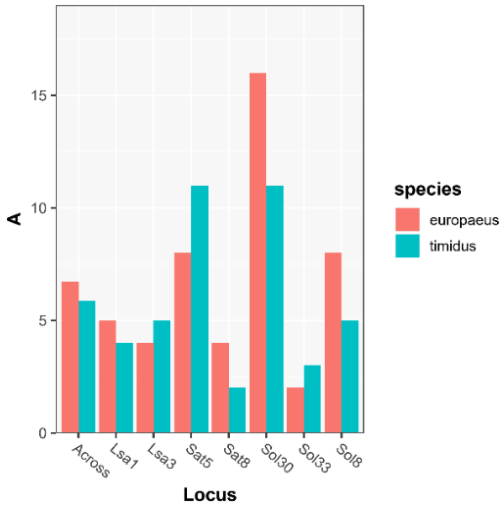

Figure 12: Mean allele counts for *L. europaeus* and *L. timidus.*

Table 13: Population genetic summary statistics obtained with CERVUS, given for each locus and across loci, for European hares and mountain hares.

L. europaeus									
Locus	**A**	**A_R**	**N**	**H_O**	**H_E**	**PIC**	**P_{ID}**	**P_{IDsib}**	**F_{NULL}**
Lsa1	5	3.13	36	0.778	0.697	0.638	0.147	0.443	0.0741
Lsa3	4	2.99	36	0.417	0.654	0.579	0.192	0.475	0.2142
Sat5	8	3.23	27	0.296	0.770	0.724	0.092	0.395	0.4407
Sat8	4	2.50	36	0.361	0.473	0.434	0.317	0.596	0.1350
Sol30	16	3.71	35	0.829	0.833	0.803	0.050	0.352	0.0173
Sol33	2	2.10	36	0.167	0.407	0.321	0.439	0.659	0.4130
Sol8	8	3.61	36	0.806	0.822	0.785	0.062	0.360	0.0030
Across Loci	6.7	3.04	36	0.522	0.665	0.612	$0.11 \cdot 10^{-5}$	0.004	
L. timidus									
Locus	**A**	**A_R**	**N**	**H_O**	**H_E**	**PIC**	**P_{ID}**	**P_{IDsib}**	**F_{NULL}**
Lsa1	4	2.54	51	0.529	0.533	0.465	0.285	0.557	0.0086
Lsa3	5	1.97	51	0.137	0.360	0.340	0.430	0.679	0.4634
Sat5	11	2.73	47	0.277	0.511	0.489	0.261	0.563	0.2929
Sat8	2	1.63	51	0.216	0.194	0.174	0.671	0.821	0.0511
Sol30	11	3.01	51	0.529	0.594	0.565	0.193	0.504	0.0574
Sol33	3	2.88	51	0.529	0.596	0.520	0.238	0.515	0.0589
Sol8	5	1.99	51	0.294	0.313	0.294	0.492	0.718	0.0698
Across Loci	5.9	2.39	51	0.359	0.443	0.407	$0.49 \cdot 10^{-3}$	0.033	

Principal Component Analysis

The PCA did not show distinct clusters for the species analyzed. Nevertheless, the representation of the PCA depicted a right shift for most *L. timidus* samples based on the first principal component (axis), and a left shift for most *L. europaeus* samples (Figure 13). Thus, the first component was able to distinguish relatively well between species. However, some *L. timidus* samples were categorized with other *L. europaeus* samples and vice versa. Additionally, *L. timidus* samples seemed to be more loosely spread than *L. europaeus* samples, which were more densely clustered at the axes displayed. The two most informative axes accounted for approximately 20% of the total variation (Figure 13).

The sample of unknown species (NA602) was depicted more closely to the *L. timidus* genotypes than the *L. europaeus* genotypes and is thus considered to be more similar to mountain hares than

European hares in the data set. One of the hybrid samples (Hb597) was shown close to the *L. timidus* samples and the other hybrid sample (Hb604) was more on the side of *L. europaeus*. In addition, Le584 was shown in the ellipse of the mountain hare samples and Lt107 was plotted in the ellipse of the other European hare samples. Consequently, some samples were shown to be more similar to the other species than the one they were identified as by the collector in the field.

The PCA using both tissue and NP samples revealed the same qualitative differences among species and samples as the PCA using only the tissue samples (Figures 13 & 14). In this result representation, European hare samples showed a left shift on the first axis compared to mountain hare samples. Hereby, two samples from the NP data set (LtNP60 & LtNP95) were estimated to be more similar to European hare samples. In this analysis, the first two components also accounted for approximately 20% of the variation in the data (Figure 14).

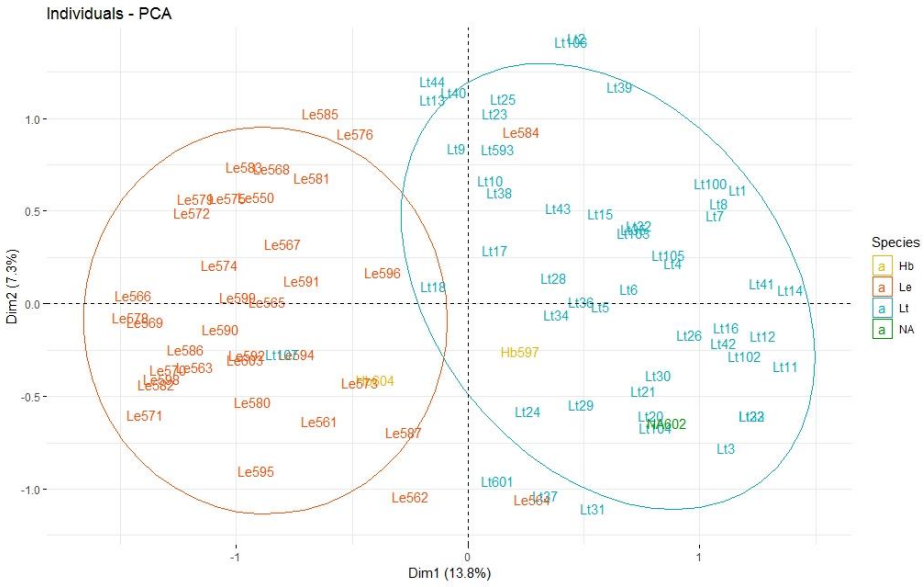

Figure 13: Results of the principal component analysis for the tissue sample genotypes. Percentage values represent variation justified by each axis.

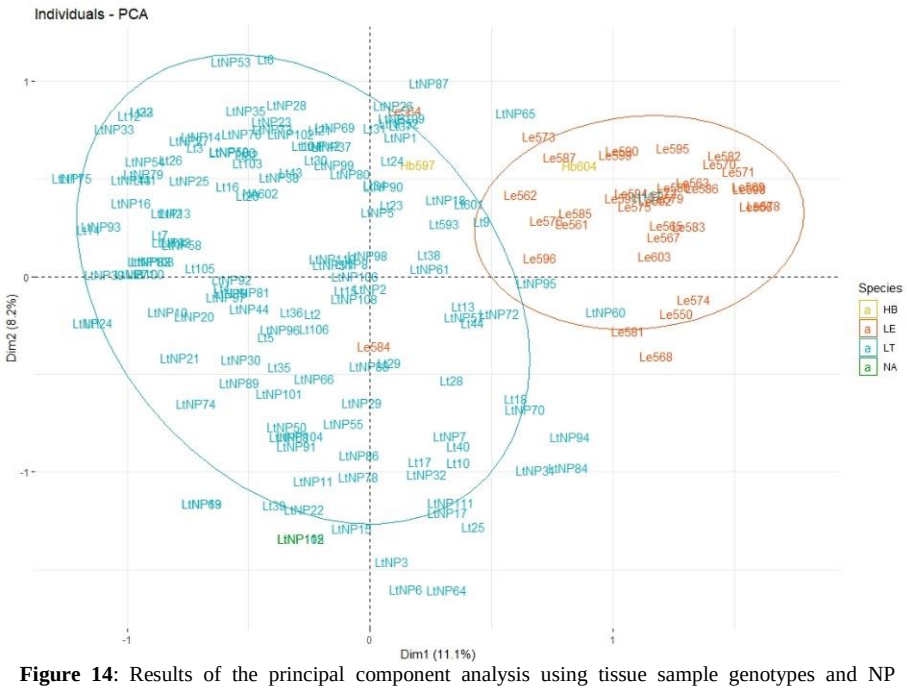

Figure 14: Results of the principal component analysis using tissue sample genotypes and NP genotypes, per-centage values represent variation justified by each axis.

STRUCTURE

STRUCTURE HARVESTER (Earl & Holdt 2012) revealed the most likely number of clusters to be two (K = 2). After merging the ten iterations from STRUCTURE in CLUMPP, the assignment probabilities indicated that some individuals previously classified as one species showed more similarities to the other species respectively contain admixture (Figure 15 & 16).

Concerning the tissue samples, Lt107 was assigned to the *L. europaeus* cluster at a high probability (P_{Le} = 0.99) and sample Lt601 was assigned to the *L. europaeus* cluster at a probability of 0.97 (Figure 15). In the PCA, Lt107 was shown to be more similar to other European hares than to mountain hares. Lt601 was depicted only slightly on the side of mountain hares (Figure 13). One of the hybrid samples (Hb 597) showed more similarity to the analyzed *L. timidus* samples (P_{Lt} = 0.93), whereas the other hybrid sample (Hb604) showed an affinity to the analyzed *L. europaeus* samples (P_{Le} = 0.78, Figure 15). These assignments were in accordance with the results obtained in the PCA. Sample Le584, which was determined to be more similar to the mountain hare samples in the PCA, did not show clear tendencies to either cluster in STRUCTURE, but contained probabilities to be assigned to both clusters (P_{Le} = 0.41). The sample of unknown species (NA602) showed a high assignment probability to the mountain hare cluster (P_{Lt} = 0.93), which was also shown with the PCA.

The analysis of the combined data set revealed individual 60 (NP60) to have a high probability of belonging to the *L. europeaus* cluster (P_{Le} = 0.97). Individual NP60 was also estimated to be more similar to the European hare samples in the PCA. Individual 60 was genotyped based on 10 samples collected in the study area in the National Park in spring 10.

In addition, some individuals from the NP data set contained admixture with European hares (Figure 16).

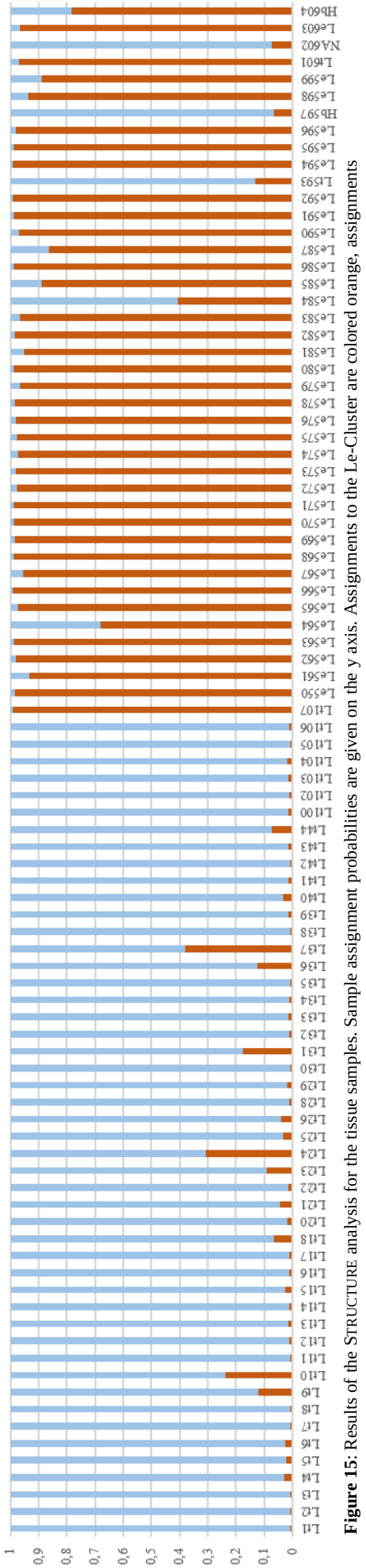

Figure 15: Results of the STRUCTURE analysis for the tissue samples. Sample assignment probabilities are given on the *y* axis. Assignments to the Le-Cluster are colored orange, assignments to the Lt-Cluster are blue.

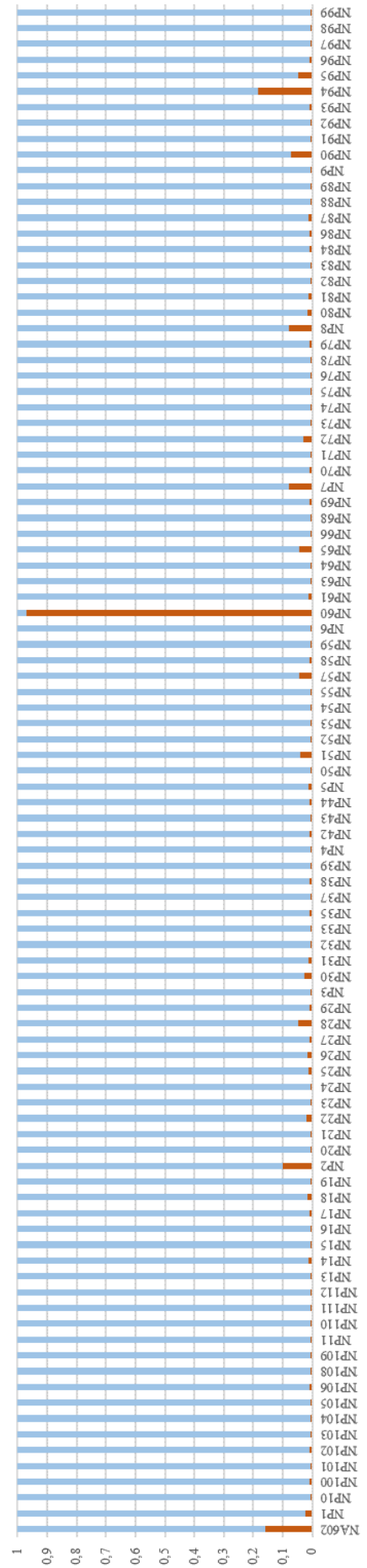

Figure 16: Results of the STRUCTURE analysis for the NP samples. The tissue samples are not displayed to reduce the amount of data in the plot.

DISCUSSION

The analysis of the molecular data gathered as part of the mountain hare monitoring in the Swiss National Park revealed between 17 and 28 individuals to have been present in the study area in each sampling session, which equals 4.86 to 8 Individuals per km^2. These estimates are higher than the values estimated by Rehnus & Bollmann (2016) based on the data of spring 2014 (3.2 ± 0.7 to 3.6 ± 1.3) and higher than the values estimated by Slotta-Bachmayr (1998) in the National Park Hohe Tauern in Austria. Nodari (2006) obtained similar values than the estimates of this study for mountain hare density (5 – 11 hares/km^2) in the Stilfser Joch National Park in Italy. In this study, in most sessions, more males than females were detected, with the estimated mean male-female ratio being 1.63 in spring and 1.11 in fall. On average, more individuals were present in the study area in spring than in fall. This difference was more pronounced for males, who were more common in spring in all study years. The average apparent seasonal survival rate was estimated to be approximately the same for females and males, however, average annual survival was higher for females than males. For both sexes, apparent survival estimates were higher from fall to spring than from spring to fall. Males showed higher fluctuations from season to season than females and additionally, males were estimated to show a higher probability for temporary migration from and into the study area. To my knowledge, no study has compared differences in mountain hare densities between seasons in the Alps. However, some studies have investigated home range sizes and have found home ranges to be smaller in fall (September – November, post breeding season) than in winter (September – March, ground covered with snow, Gamboni et al. (2008), Nodari (2006)). Additionally, feces density was observed to be smaller in winter, when the ground was covered in snow, than in summer, and consequently more individuals were assumed to be present in summer than in winter (Rehnus et al. 2013). In contrast to these findings, more samples, more individuals and more samples per individual were found in spring (March/April) than in fall (October) in this study. However, feces density could also increase due to higher activity of hares and more nutrition being available (Rehnus et al. 2013).

The discrimination of mountain and European hares was demonstrated by some differences, which included allele frequencies and heterozygosity values. The application of these findings to the monitoring data from the National Park implied a European hare to have been present in the study area in spring 2016.

The number of individuals in the National Park was estimated based on multilocus genotypes at seven loci and a qualitative integration of two additional loci. Multilocus genotype groups were assigned to the same individual when they did not show more than two allele mismatches across all nine markers. Thus, animals found to have the same multilocus genotype were classified as the same individual. However, this is not necessarily always the case, because the probability exists that two random individuals share a common genotype (Mills et al. 2000), which may lead to misidentifications of individuals. Misidentifications in CMR monitoring may lead to an overestimation of survival and an underestimation of abundance, but can greatly be overcome by using a sufficient number of high resolution microsatellites resulting in a high power to identify individuals (Lukacs & Burnham 2005). The power of a set of markers to distinguish relationships and individuals depends on the number of markers, the allele frequency distribution and the typing error rate of each marker (Wang 2006).

One possible estimation of the power of markers is given by the probability of identity, which describes the proportion of random individuals that share the same genotype by chance (Waits & Leberg 2000). Additionally, the probability of identity may be described under the assumption of the individuals in the sample being related (Waits et al. 2001). Woods et al. (1999) apply a P_{IDsib} of 0.05 for the individual-based analysis of brown and black bears as the highest acceptable value for individual identification. Here, the obtained P_{IDsib} value (0.036) lies below this recommendation. Nevertheless, the value is still relatively high and thus identifying individuals with the markers

applied here needs to be done with caution. However, to identify unique genotypes from samples, two additional loci were used. The identification of unique genotypes was consequently done with additional information not considered in the calculation of P_{IDsib}. Considering these loci in the calculation of P_{IDsib} would lower its estimated value across loci.

To conduct the pedigree analysis, the two additional loci could not be taken into consideration either. Thus, the calculated P_{IDsib} value represents the genotype data used in the pedigree reconstruction. The application of the additional loci in the process of identifying individuals led to additional identifications of individuals, compared to the application of only seven loci. More explicitly, some individuals were identical regarding the seven primarily used loci but differed at the additional loci. If the two additional loci could have been scored as bi-allelic markers, the individuals' multilocus genotypes would show all markers used for individual identification. Thus, the diversity assessed would be represented better and the pedigree reconstructed would be more reliable. Generally, a possible reason for low genetic diversity at the assessed markers would be high relatedness among individuals. However, P_{IDsib} values for mountain hares were similar based on the NP genotypes and the tissue genotypes, even though mountain hare individuals in the tissue data set originated from different locations across the Alps. Thus, high relatedness among samples in the NP data set being the only reason for the samples to show low genetic diversity at the loci assessed can be excluded.

Another possible explanation for the seven markers to show low diversity could be that the markers applied here show a limited power to distinguish individuals and assess relatedness in mountain hares in general. Some of the markers have been applied to mountain hares in other studies in different constellations (Beugin et al. 2017, Hamill et al. 2006, Thulin et al. 2006a). However, for these 7 loci, the power of the markers to identify European hares was found to be higher than for mountain hares. For example, the P_{IDsib} value estimated for *L. europaeus* was tenfold lower than for the mountain hares. This implies a lower probability of multilocus genotypes of related individuals being the same by chance. The lower P_{IDsib} value further indicates that the diversity of the alleles found at these markers is larger for European hares than for mountain hares.

These results are also reflected in the values obtained for heterozygosity and the allele counts for both species. At locus Sat5, the difference of expected and observed heterozygosity was larger for *L. europaeus* than for *L. timidus* samples. In general, the locus specific expected heterozygosity (H_E) is calculated as follows:

$$H_E = 1 - \sum_{i}^{n} p_i^2$$

whereby p_i equals the frequency of the i-th allele and n stands for the number of alleles at the locus. Thus, the expected frequency depends on the frequency of each allele at the locus and the number of alleles found. For locus Sat5, allele counts differ only slightly between species, but the frequencies of the alleles vary: For mountain hares, allele 198 is by far the most common, with all other alleles accounting for only small frequencies. For European hares, the frequencies are distributed more evenly: Allele 198 is less common, and instead other alleles are more common (e. g. allele 221). Due to expected heterozygosity values increasing with more even distributions of allele frequencies, the expected heterozygosity value for *L. europaeus* at locus Sat5 is larger than for *L. timidus*.

However, expected heterozygosity is calculated based on the assumption that the populations are at Hardy-Weinberg equilibrium (Hardy 1908, Weinberg 1908). However, samples of each species were collected across different locations in the Alps and could thus originate from separate, independent populations. Thus, it is possible that the assumptions of Hardy-Weinberg are not met. If this were the case, expected frequency calculations would be unreliable. To test if Hardy-Weinberg assumptions are met, population differentiation would have to be assessed using genetic

differentiation measures (e.g. F statistics) or exact tests for deviation of Hardy-Weinberg equilibrium (Fryxell et al. 2014, Hamill et al. 2006, Hamilton 2009, Holderegger & Wagner 2008). Such measures compare allele frequencies across subpopulations and assess differences or compare expected and observed heterozygosity values (Hamilton 2009). However, these calculations would not deliver reliable results using the tissue genotypes from this study because (i) no clear subpopulations can be defined, as exact locations of the samples' origins are not always known, (ii) the number of samples from different locations show a high range in values, and (iii) the sample number was generally too low for each species.

The comparison of *L. timidus* and *L. europaeus* revealed some differences in allele frequencies, allele counts as well as heterozygosity levels. These differences were assessed using STRUCTURE as well as through the application of a PCA. Both multivariate methods indicated some differences between European and mountain hares. In both methods, the same mountain hare individuals were estimated to belong to the European hare cluster and vice versa. For example, the PCA suggested Lt107 (*L. timidus*) to be more similar to European hares. STRUCTURE analysis confirmed this result and suggested Lt107 to belong to the European hare cluster with a probability of 0.97. Additionally, the sample of undefined species (NA602) was estimated to originate from a mountain hare by the PCA as well as STRUCTURE ($P_{Lt} = 0.93$). Using a combination of the the tissue sample genotype data and the NP genotypes, one individual detected in the in the National Park was identified as European hare, with a high assignment probability in STRUCTURE ($P_{Le} = 0.97$) and by the PCA. The individual (male, IndID 60) was found to have been present in the park during spring 2016 and identified based on 10 samples. The individual was not detected in any subsequent sampling sessions. Further confirmation of the species assignment of individual 60 was obtained using mtDNA sequencing, which confirmed the results obtained in this study. MtDNA sequences differ between mountain and European hares and can thus be used for reliable species identification (Thulin 2003, Thulin et al. 2006b). Additionally, samples gathered noninvasive often contain a higher amount of mtDNA than nuclear DNA and mtDNA sequencing consequently often delivers reliable results (Waits & Paetkau 2005). However, first generation hybrids would not be detected, as hybridization between mountain and European hares is unidirectional.

The values for observed heterozygosity do not differ between species and among species between tissue and NP sample genotypes. Concerning mountain hares, a homozygous excess was observed in the tissue as well as the NP sample genotypes at loci Lsa3 and Sat5. An excess of homozygote genotypes may be explained by various factors. First, it may be due to a few alleles being highly common, which may happen by chance (Hamilton 2009). Second, inbreeding in a population may cause Hardy-Weinberg assumptions to be violated and a higher than expected homozygote frequency (Hamilton 2009). Inbreeding is mostly quantified through the calculation of an inbreeding coefficient, which is based on pedigree analysis (Hamilton 2009). As in this study no pedigree could be reconstructed reliably, an estimation of inbreeding was not possible. Third, technical errors may lead to some alleles being missed due to high error rates (Pompanon et al. 2005). This was attempted to be assessed by quantifying error rates in the data set. Fourth, homozygous excess may be due to the presence of null alleles. Null alleles are alleles which fail to amplify during the PCR (Dakin & Avise 2004). The presence of null alleles leads to either an overabundance of homozygotes or a high number of allelic dropouts, if the individual would have been homozygous for the null allele (Wagner et al. 2006). The presence of null alleles may lead to false parentage-exclusions in pedigree reconstructions (Wagner et al. 2006). However, the degree of bias null alleles cause strongly depends on their frequencies (Dakin & Avise 2004). Incorporating proper error models in pedigree analysis, such as the likelihood method implemented in COLONY (Wang 2004, Wang and Santure 2009), may decrease the seriousness of this problem and highly accurate sibship and parentage assignments may be obtained even when markers with high error rates are used (Wang 2019). Many markers are needed to obtain reasonably good parentage and sibship assignments when the rate of false alleles is high, and when only a few

markers are applied, a moderate rate of false alleles can still cause substantial loss of inference accuracy (Wang 2019). In this study, using higher error rates for pedigree reconstruction led to different results than applying lower error rates. Only a few sib-ship clusters were found to be constant for both error estimates. A crucial step when integrating error models in pedigree analysis is the quantification of the error rate. Here, error rates were obtained based on the proportion of false homozygotes in heterozygote genotypes as well as the proportion of missing replicates in the replicates produced using the multitube approach. The quantification of errors based on missing replicates does not necessarily show the probability of error in the final consensus genotype. Thus, other quantifications to more reliably describe errors in the consensus multilocus genotypes would have to be integrated in the pedigree analysis (Pompanon et al. 2005).

To reduce errors in the obtained consensus multilocus genotype, a modified multitube approach (Taberlet et al. 1996) was implemented. Here, only three PCRs were performed per sample. A homozygote replicate was not accepted until it had been confirmed three times and no allele was accepted until it was at least detected twice. The strict rule for detecting homozygotes was applied to ensure a minimum allelic dropout and increase the probability of consensus homozygous genotypes to be true homozygotes: A heterozygote is falsely scored as a homozygote if one allele drops out and the likelihood of dropping out in all PCRs decreases with more replicates (Taberlet et al. 1996). Assuming a maximum dropout rate of 0.5 (Taberlet et al. 1996), a triple dropout has a probability of 0.5^3, which results in a probability of 0.125. However, these results are based on simulations and as a large variety of mean dropout rates have been reported (Broquet et al. 2007, Broquet & Petit 2004, Creel et al. 2003), it is important to quantify study specific dropout and error rates.

Here, error rates (missing genotypes and false homozygotes) did not differ significantly between loci considering means of all years. The highest rate of missing genotypes was observed at locus Lsa3. However, comparisons between loci need to be done with care, as loci show different amounts of heterozygosity. Loci showing higher amounts of heterozygosity are bound to show higher amounts of false homozygotes. Additionally, consensus homozygote replicates can only contain complete replicate dropouts, as single allele dropouts automatically lead to the whole genotype dropping out. Thus, the more consensus homozygote replicates and the less heterozygosity at a locus, the less likely it is to detect false homozygotes and the more likely it is to detect missing replicates.

Further, errors were observed during the determination of sex. Some samples classified as females showed matches to groups of male genotypes and were consequently classified as false females (2%, N = 12). Male genotypes showing matches to female genotype groups were reanalyzed with the two additional loci and always found to be different. Thus, no false male genotypes were observed in the data set. If the match of false females to male genotypes would be assumed to be correct, it would imply that 2% of all female samples have been genotyped as the wrong sex. A certain error rate in sex determination using Sry was also detected by Wallner et al. (2001). Nevertheless, the possibility exists that these samples share a common genotype while not originating from the same individual, which is supported by a relatively large P_{IDsib} value. If these individuals would be assumed to be females, this would contribute to a lower sex ratio, and a lower number of samples per individual for females.

The classification of these individuals as unknown sex led a slightly higher mean number of samples collected for females than for males. Generally, different factors are thought to influence the number of samples collected and how many individuals are identified with the respective number of samples. First, the number of species present in the area defines how many feces are available for collection. Second, the researcher needs to find the samples, which may be easier or more difficult depending on the conditions in the area. In the spring sampling session, snow was mostly covering the study area and feces were easy to spot. In fall sessions, the ground was usually

covered with grass, moss or rocks, which made the detection of feces more difficult. However, Rehnus conducted an experiment, where he found that most of the samples in the area are also detected (personal communication). In addition, more samples were found through systematic as well as opportunistic sampling. If more samples would have been collected only because samples are spotted more easily, the difference in the number of samples collected between seasons for systematic sampling would less severe.

However, the quality of samples may differ depending on the conditions in the study area. If snow is covering the area, samples are automatically preserved by the cooler temperatures as well as the snow itself. Thus, the quality and quantity of the DNA in the sample may be higher. This is supported by the result of the amount of missing data in the obtained replicates being significantly smaller in spring than in fall. However, not only the number of genotyped samples but also the number of collected samples was higher in spring than in fall.

Whether a sample was collected for an individual in a sampling session led to a capture history of each individual. This was then used to estimate apparent survival rates and rates for temporary immigration and emigration in MARK. Males were found to have a more fluctuating survival rate and higher rates for temporary immigration and emigration. Females' survival rate was found to be more constant throughout time. However, considering the average survival rate across all years, the survival of both sexes was similar. The survival rate estimates were based on whether or not an individual was observed in the study area. Thus, if no sample was collected for an individual, the individual was assumed to be either temporarily or permanently absent from the study area. However, just because no sample was genotyped, it does not necessarily mean that the individual was absent. First, high dropout rates may cause missing genotypes, which leads to not observing an individual if only one sample was found in the sampling session. Consequently, high allelic dropout rates may cause an underestimation of individual abundance.

However, the number of individuals identified does not solely depend on the number of samples collected, as there is no direct linear correlation between the number of samples and the number of individuals identified (Appendix 3). Generally, more individuals were identified with more samples. However, at a certain number of samples, no additional individuals were identified with additional samples. For example, in fall 2018, fall 2016 and spring 2014, 25 individuals were detected in the study area. Nevertheless, 163 samples were collected in fall 2018 and in fall 2016 only 103 samples were collected. Thus, the collection of additional 60 samples did not lead to any additional individual identifications. Consequently, it is assumed that a minimum number of samples are necessary to identify most individuals present, but it is not necessary to maximize sample collection.

CONCLUSIONS

Through the application of seven nucleotide markers, 59 male and 31 female individuals were identified in the study area in the Swiss National Park and it could be shown that there are more males than females present, even if males show higher probabilities for migration and lower apparent annual survival rates. Additionally, some differences between European and mountain hares were shown for the seven nuclear markers applied in the NGS monitoring. The results of the analysis done within the framework of this master thesis highlight the necessity of using an adequate number of high-quality microsatellite markers to establish pedigrees and identify hybrids in noninvasive genetic monitoring. However, for the identification of species I suggest analyzing mtDNA sequences, as this analysis gives more exact results for species assignments and the quality of mtDNA in samples gathered noninvasively is higher than that of nuclear DNA. Through a combination of the application of high-quality nucleotide markers and mtDNA sequence analysis, it would be possible to identify species, reconstruct pedigrees and analyze population abundance.

ACKNOWLEDGEMENTS

I thank Felix Gugerli, Kurt Bollmann, Maik Rehnus and Rolf Holderegger for their relentless support, their constructive critics, and their help with various issues during this thesis. Further, I want to thank Sabine Brodbeck for her continuous help in the lab and for support in diverse parts of this thesis. Lastly, I also want to thank the whole ecological genetics group at the WSL for always being available to answer questions. Thank you!

REFERENCES

Acevedo, P., A. Jimenez-Valverde, J. Melo-Ferreira, R. Real, and P. C. Alves. 2012. Parapatric species and the implications for climate change studies: a case study on hares in Europe. Global Change Biology **18**:1509-1519.

BAFU. 2018. Eidgenössische Jagdstatistik. Wildtier Schweiz. https://www.uzh.ch/wild/ssl-dir/jagdstatistik/?page=wildtiere&dt=1&gr=2&tr=30&th=1&ca=CH&co=CH&caco=1&ys=1933&ye=2017&lang=de. Accessed on: 04.03.2019.

Beever, E. A., C. Ray, J. L. Wilkening, P. F. Brussard, and P. W. Mote. 2011. Contemporary climate change alters the pace and drivers of extinction. Global Change Biology **17**:2054-2070.

Beja-Pereira, A., R. Oliveira, P. C. Alves, M. K. Schwartz, and G. Luikart. 2009. Advancing ecological understanding through technological transformations in noninvasive genetics. Molecular Ecology Resources **9**:1279-1301.

Beugin, M.-P., J. Letty, C. Kaerle, J.-S. Guitton, L. Muselet, G. Queney, and D. Pontier. 2017. A single multiplex of twelve microsatellite markers for the simultaneous study of the brown hare (*Lepus europaeus*) and the mountain hare (*Lepus timidus*). Ecology and Evolution **7**:3931-3939.

Bisi, F., L. A. Wauters, D. G. Preatoni, and A. Martinoli. 2015. Interspecific competition mediated by climate change: which interaction between brown and mountain hare in the Alps? Mammalian Biology **80**:424-430.

Botstein, D., R. L. White, M. Skolnick, and R. W. Davis. 1980. Construction of a genetic linkage map in man using restriction fragment length polymorphisms. American Journal of Human Genetics **32**:314-331.

Broquet, T., N. Menard, and E. J. Petit. 2007. Noninvasive population genetics: a review of sample source, diet, fragment length and microsatellite motif effects on amplification success and genotyping error rates. Conservation Genetics **8**:249-260.

Broquet, T., and E. J. Petit. 2004. Quantifying genotyping errors in noninvasive population genetics. Molecular Ecology **13**:3601-3608.

Burnham, K. P., and D. R. Anderson. 2002. Model selection and multimodel inference: a practical information-theoretic approach. Springer-Verlag, New York.

Cattell, R. 1966. The scree test for the number of factors. Multivariate Behavioral Research **1**:245-276.

Creel, S., and E. Rosenblatt. 2013. Using pedigree reconstruction to estimate population size: genotypes are more than individually unique marks. Ecology and Evolution **3**:1294-1304.

Creel, S., G. Spong, J. L. Sands, J. Rotella, J. Zeigle, L. Joe, K. M. Murphy, and D. Smith. 2003. Population size estimation in Yellowstone wolves with error-prone noninvasive microsatellite genotypes. Molecular Ecology **12**:2003-2009.

Dahl, F., and T. Willebrand. 2005. Natal dispersal, adult home ranges and site fidelity of mountain hares *Lepus timidus* in the boreal forest of Sweden. Wildlife Biology **11**:309-317.

Dakin, E. E., and J. C. Avise. 2004. Microsatellite null alleles in parentage analysis. Heredity **93**:504-509.

Dray, S., and A.-B. Dufour. 2007. The ade4 Package: Implementing the duality diagram for ecologists. Journal of Statistical Software **22**.

Earl, D. A., and B. M. v. Holdt. 2012. STRUCTURE HARVESTER: a website and program for visualizing STRUCTURE output and implementing the Evanno method. Conservation Genetics Resources **4**:359-361.

El-Mousadik, A., and R. J. Petit. 1996. High level of genetic differentiation for allelic richness among populations of the argan tree (*Argania spinosa* (L.) Skeels) endemic to Morocco. Theoretical and Applied Genetics **92**:832-839.

Evanno, G., S. Regnaut, and J. Goudet. 2005. Detecting the number of clusters of individuals using the software STRUCTURE: a simulation study. Molecular Ecology **14**:2611-2620.

Fryxell, J. M., A. R. E. Sinclair, and G. Caughly. 2014. Wildlife Ecology, Conservation, and Management. John Wiley & Sons Ltd, Oxford.

Galpern, P., M. Manseau, P. Hettinga, K. Smith, and P. Wilson. 2012. Allelematch: an R package for identifying unique multilocus genotypes where genotyping error and missing data may be present. Molecular Ecology Resources **12**:771-778.

Gamboni, A.-S. G., F. Bisi, E. Masseroni, M. Nodari, D. G. Preatoni, L. A. Wauters, A. Martinoli, and G. Tosi. 2008. Home range dynamics of mountain hares (*Lepus timidus*) in the Swiss Alps. Hystrix-Italian Journal of Mammalogy **19**:77-84.

Giger, R. Schneehase in den Alpen. Eidgenössisches Forschungsinstitut für Wald, Schnee und Landschaft (WSL), https://www.wsl-junior.ch/fileadmin/user_upload/WSL-Junior/News/Hilfe_fuer_Schneehasen/Schreibkarte_Schneehase_2017_gzd2_01.jpg.

Gruber, B., and A. T. Adamack. 2014. PopGenReport: Simplifying basic population genetic analysis in R. Methods in Ecology and Evolution **5**:384-387.

Hamill, R. M., D. Doyle, and E. J. Duke. 2006. Spatial patterns of genetic diversity across European subspecies of the mountain hare, *Lepus timidus* L. Heredity **97**:355-365.

Hamilton, M. B. 2009. Population Genetics. John Wiley & Sons Ltd, West Sussex, UK.

Hardy, G. H. 1908. Mendelian proportions in a mixed population. Science **28**:41-50.

Hearne, C. M., S. Ghosh, and J. A. Todd. 1992. Microsatellites for linkage analysis of genetic traits. Trends in Ecology and Evolution **8**:288-294.

Holderegger, R., and H. H. Wagner. 2008. Landscape genetics. Bioscience **58**:199-207.

Höss, M., M. Kohn, S. Pääbo, F. Knauer, and W. Schröder. 1992. Excrement analysis by PCR. Nature **359**:199-199.

Hughes, L. 2000. Biological consequences of global warming: is the signal already apparent? Trends in Ecology & Evolution **15**:56-61.

Jacob, G., R. Debrunner, F. Gugerli, B. Schmid, and K. Bollmann. 2010. Field surveys of capercaillie (*Tetra urogallus*) in the Swiss Alps underestimated local abundance of the species revealed by genetic analyses of non-invasive samples. Conservation Genetics **11**:33-44.

Jakobsson, M., and N. A. Rosenberg. 2007. CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. Bioinformatics **23**:1801-1806.

Jolly, G. M. 1965. Explicit estimates from capture-recapture data with both death and immigration - stochastic models. Biometrics **50**:88-97.

Jombart, T. 2008. adegenet: a R package for the multivariate analysis of genetic markers. Bioinformatics **24**:1403-1405.

Jones, O. R., and J. Wang. 2010. COLONY: a program for parentage and sibship inference from multilocus genotype data. Molecular Ecology Resources **10**:551-555.

Kalinowski, S. T., M. L. Taper, and T. C. Marshall. 2007. Revising how the computer program CERVUS accommodates genotyping error increases success in paternity assignment. Molecular Ecology **16**:1099-1106.

Kassambara, A., and F. Mundt. 2017. factoextra: Extract and visualize the results of multivariate data analysis. https://CRAN.R-project.org/package=factoextra.

Kendall, W. L., J. D. Nichols, and J. E. Hines. 1997. Estimating temporary emigration using capture-recapture data with Pollock's robust design. Ecology **78**:563-578.

Kormann, U., F. Gugerli, N. Ray, L. Excoffer, and K. Bollmann. 2012. Parsimony-based pedigree analysis and individual-based landscape genetics suggest topography to restrict dispersal and connectivity in the endangered capercaillie. Biological Conservation **152**:241-252.

Kryger, U. 2002. Genetic variation among South African hares (*Lepus* spec.) as inferred from mitochondrial DNA and microsatellites. PhD Thesis. University of Pretoria, Pretoria.

Lampa, S., K. Henle, R. Klenke, and B. Gruber. 2015. How to overcome genotyping errors in non-invasive genetic mark-recapture population size estimation - a review of available methods illustrated by a case study. The Journal of Wildlife Management **77**:1490-1511.

Leach, A. C. K., F. Santilli, J. Rintala, P. Helle, J. Tiainen, F. Bisi, A. Martinoli, W. I. Montgomery, and N. Reid. 2016. Niche overlap of mountain hare subspecies and the vulnerability of their ranges to invasion by the European hare; the (bad) luck of the Irish. Biological Invasions.

Lönneberg, E. 1905. On hybrids between *Lepus timidus L.* and *Lepus europaeus Pall.* from southern Sweden. Proceedings of Zoological Society of London **1**:278-287.

Luikart, G., N. Ryman, D. A. Tallmon, M. K. Schwartz, and F. W. Allendorf. 2010. Estimating of census and effective population sizes: the increasing usefulness of DNA-based approaches. Conservation Genetics **11**:355-373.

Lukacs, P. M., and K. P. Burnham. 2005. Review of capture-recapture methods applicable to noninvasive genetic sampling. Molecular Ecology **14**:3909-3919.

Mendiburu, F. d. 2019. agricolae: Statistical procedures for agricultural research. La Molina, Peru.

Mills, L. S. 2013. Wildlife Populations: Demography, Genetics, and Management. John Wiley & Sons, West Sussex, UK.

Mills, L. S., J. J. Citta, K. P. Lair, M. K. Schwartz, and D. A. Tallmon. 2000. Estimating animal abundance using noninvasive DNA sampling: Promise and pitfalls. Ecological Applications **10**:283-294.

Moritz, C., J. L. Patton, C. J. Conroy, J. L. Parra, G. C. White, and S. R. Beissinger. 2008. Impact of a century of climate change on small-mammal communities in Yosemite National Park, USA. Science **322**:261-264.

Mougel, F., J.-C. Mounolou, and M. Monnerot. 1997. Nine Polymorphic microsatellite loci in the rabbit, *Oryctolagus cuniculus*. Animal Genetics **28**:58-71.

Newey, S., F. Dahl, T. Willebrand, and S. Thirgood. 2007. Unstable dynamics and population limitation in mountain hares. Biological Reviews **82**:527-549.

Nodari, M. 2006. Ecological role of mountain hare (*Lepus timidus*) in the alpine ecosystem. Habitat use, population consistency and dynamics of a species of conservation and management interest. Dissertation. University Insubria.

Parmesan, C. 2006. Ecological and evolutionary responses to recent climate change. Annual Review of Ecology Evolution and Systematics **37**:637-669.

Pehrson, A., and B. Lindlöf. 1984. Impact of winter nutrition on reproduction in captive Mountain hares (*Lepus timidus*) (Mammalia: Lagomorpha). Journal of Zoology **204**:201-209.

Piggott, M. P., and A. C. Taylor. 2003. Remote collection of animal DNA and its applications in conservation management and understanding the population biology of rare and cryptic species. Wildlife Research **30**:1-13.

Pollock, K. H. 1982. A capture-recapture design robust to unequal probability of capture. Journal of Wildlife Management **46**:752-757.

Pollock, K. H. 2000. Capture-Recapture Models. Journal of the American Statistical Association **95**:293-296.

Pompanon, F., A. Bonin, E. Bellemain, and P. Taberlet. 2005. Genotyping errors: Causes, consequences and solutions. Nature Reviews Genetics **6**:847-859.

Porras-Hurtado, L., Y. Ruiz, C. Santos, C. Phillips, A. Carracedo, and M. V. Lareu. 2013. An overview of Structure: Applications, parameter settings, and supporting software. Frontiers in Genetics **4**:98-98.

Pradel, R. 1996. Utilization of capture-mark-recapture for the study of recruitment and population growth rate. Biometrics **52**:703-709.

Pritchard, J. K., M. Stephens, and P. Donnelly. 2000. Inference of population structure using multilocus genotype data. Genetics **155**:945-959.

R Core Team. 2018. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Rehnus, M. 2013. Der Schneehase in den Alpen. Ein Überlebenskünstler mit ungewisser Zukunft. Zürich, Bristol-Stiftung; Bern, Haupt.

Rehnus, M., and K. Bollmann. 2016. Non-invasive genetic population density estimation of mountain hares (*Lepus timidus*) in the Alps: systematic or opportunistic sampling? European Journal Wildlife Research **62**:737-747.

Rehnus, M., K. Bollmann, D. R. Schmatz, K. Hackländer, and V. Braunisch. 2018. Alpine glacial relict species losing out to climate change: The case of the fragmented mountain hare population (*Lepus timidus*) in the Alps. Global Change Biology **24**:3236-3253.

Rehnus, M., L. Marconi, K. Hackländer, and F. Filli. 2013. Seasonal changes in habitat use and feeding strategy of the mountain hare (*Lepus timidus*) in the Central Alps. Hystrix-Italian Journal of Mammalogy **24**:161-165.

Rehnus, M., M. Wehrle, and R. Palme. 2014. Mountain hares *Lepus timidus* and tourism: stress events and reactions. Journal of Applied Ecology **51**:6-12.

Rico, C., I. Rico, N. M. Webb, S. A. Smith, D. Bell, and G. Hewitt. 1994. Four polymorphic microsatellite loci for the European wild rabbit, *Oryctolagus cuniculus*. Animal Genetics **25**:367-367.

Roedenbeck, I. A., and P. Voser. 2008. Effects of roads on spatial distribution, abundance and mortality of brown hare (*Lepus europaeus*) in Switzerland. European Journal of Wildlife Research **54**:425-437.

Rosner, S., R. Brandl, G. Segelbacher, T. Lorenc, and J. Muller. 2014. Noninvasive genetic sampling allows estimation of capercaillie numbers and population structure in the Bohemian Forest. European Journal of Wildlife Research **60**:789-801.

Sawaya, M. A., J. B. Stetz, A. P. Clevenger, M. L. Gibeau, and S. T. Kalinowski. 2012. Estimating grizzly and black bear population abundance and trend in Banff National Park using noninvasve genetic sampling. Plos One **7**:e34777.

Schwarz, C. J., and G. A. F. Seber. 1999. Estimating Animal Abundance: Review III. Statistical Science **14**:427-456.

Seber, G. A. F. 1965. A note on the multiple recapture census. Biometrika **52**:249-259.

Silvy, N. J., R. R. Lopez, and M. J. Peterson. 2012. Techniques for marking wildlife. Pages 230-257 The wildlife techniques manual: Research Johns Hopkins University Press.

Sloan, M. A., P. Sunnucks, D. Alpers, B. Behregaray, and A. C. Taylor. 2000. Highly reliable genetic identification of individual northern hairy-nosed wombats from single remotely collected hairs: a feasible censusing method. Molecular Ecology **9**:1233-1240.

Slotta-Bachmayr, L. 1998. Biologie und Ökologie des Alpenschneehasen (*Lepus timidus varronis* Miller 1901). Verbreitung, Raumnutzung, Aktivität und Habitatwahl in den Hohen Tauern. . Paris Lodron University, Salzburg.

Smith, A. T., and C. H. Johnston. 2008a. *Lepus europaeus*. http://dx.doi.org/10.2305/IUCN.UK.2008.RLTS.T41280A10430693.en. Accessed on: 04.03.2019.

Smith, A. T., and C. H. Johnston. 2008b. *Lepus timidus*. IUCN. http://dx.doi.org/10.2305/IUCN.UK.2008.RLTS.T11791A3306541.en. Accessed on: 04.03.2019.

Smith, R. K., N. V. Jennings, and S. Harris. 2005. A quantitative analysis of the abundance and demography of European hares *Lepus europaeus* in relation to habitat type, intensity of agriculture and climate. Mammal Review **35**:1-24.

Spitzenberger, F. 2001. *Lepus Europaeus*. Pages 317-324 Die Säugetierfauna Österreichs. Bundesministerium für Land- und Forstwirtschaft, Graz.

Stenglein, J. L., L. P. Waits, D. E. Ausband, P. Zager, and C. M. Mack. 2010. Efficient, noninvasive genetic sampling for monitoring reintroduced wolves. The Journal of Wildlife Management **74**:1050-1058.

Summers, K., and W. Amos. 1997. Behavioral, ecological, and molecular genetic analyses of reproductive strategies in the Amazonian dart-poison frog, *Dendrobates ventrimaculatus*. Behavioral Ecology **8**:260-267.

Surrige, A. K., D. J. Bell, C. Rico, and G. M. Hewitt. 1997. Polymorphic microsatellite loci in the European rabbit (*Oryctolagus cuniculus*) are also amplified in other lagomorph species. Animal Genetics **28**:302-305.

Taberlet, P., and J. Bouvet. 1992. Bear conservation genetics. Nature **358**:197-197.

Taberlet, P., J. J. Camarra, S. Griffin, E. Uhres, O. Hanotte, L. P. Waits, C. Dubois-Paganon, T. Burke, and J. Bouvet. 1997. Noninvasive genetic tracking of the endangered Pyrenean brown bear population. Molecular Ecology **6**:869-876.

Taberlet, P., S. Griffin, B. Goossens, S. Questiau, V. Manceau, N. Escaravage, L. P. Waits, and J. Bouvet. 1996. Reliable genotyping of samples with very low DNA quantities using PCR. Nucleic Acids Research **24**:3189-3194.

Taberlet, P., L. P. Waits, and G. Luikart. 1999. Noninvasive genetic sampling: look before you leap. TREE **14**:323-327.

Thulin, C.-G. 2003. The distribution of mountain hares *Lepus timidus* in Europe: a challenge from brown hares *L. europaeus*? Mammal Review **33**:29-42.

Thulin, C.-G., M. Fang, and A. O. Averianov. 2006a. Introgression from *Lepus europaeus* to *L. timidus* in Russia revealed by mitochondrial single nucleotide polymorphisms and nuclear microsatellites. Hereditas **143**:68-76.

Thulin, C.-G., and J. E. C. Flux. 2003. *Lepus timidus* Linnaeus, 1758 - Schneehase. Pages 155-185 Handbuch der Säugetiere Europas, Band 3/II: Hasenartige Lagomorpha. Aula Verlag, Wiebelsheim.

Thulin, C.-G., J. Stone, H. Tegelstrom, and C. W. Walker. 2006b. Species assignment and hybrid identification among Scandinavian hares *Lepus europaeus* and *L. timidus*. Wildlife Biology **12**:29-38.

Thulin, C.-G., and H. Tegelström. 2002. Biased geographical distribution of mitochondrial DNA that passed the species barrier from mountain hares to brown hares (genus Lepus): An effect of genetic incompatibility and mating behaviour? Journal of Zoology **258**:299-306.

Wagner, A. P., S. Creel, and S. Kalinowski. 2006. Estimating relatedness and relationships using microsatellite loci with null alleles. Heredity **97**:336-345.

Waits, J. L., and P. L. Leberg. 2000. Biases associated with population estimation using molecular tagging. Animal Conservation **3**:191-199.

Waits, L. P., G. Luikart, and P. Taberlet. 2001. Estimating the probability of identity among genotypes in natural populations: cautions and guidelines. Molecular Ecology **10**:249-256.

Waits, L. P., and D. Paetkau. 2005. Noninvasive genetic sampling tools for wildife biologists: A review of applications and recommendations for accurate data collection. Journal of Wildlife Management **69**:1419-1433.

Wallner, B., S. Huber, and R. Achmann. 2001. Non-invasive PCR sexing of rabbits (*Oryctolagus cuniculus*) and hares (*Lepus europeaus*). Mammalian Biology **66**:190-192.

Wang, J. 2004. Sibship reconstruction from genetic data with typing errors. Genetics **166**:1963-1979.

Wang, J. 2006. Informativeness of genetic markers for pairwise relationship and relatedness inference. Theoretical Population Biology **70**:300-321.

Wang, J. 2019. Pedigree reconstruction from poor quality genotype data. Heredity.

Wang, J., and A. W. Santure. 2009. Parentage and sibship inference from multilocus genotype data under polygamy. Genetics **181**:1579-1594.

Weinberg, W. 1908. On the demonstration of heredity in man (translated by SH Boyer 1963 in Papers on Human Genetics), Prentice-Hall Englewood Cliffs, NJ.

White, G. C., and K. P. Burnham. 1999. Program MARK: survival estimation from populations of marked animals. Bird Study **46**:120-139.

Wickham, H. 2016. ggplot2: elegant graphics for data analysis. Springer-Verlag, New York.

Woods, J. G., D. Paetkau, D. Lewis, B. N. McLellan, M. Proctor, and C. Strobeck. 1999. Genetic tagging of free-ranging black and brown bears. Wildlife Society Bulletin **27**:616-627.

APPENDIX

Appendix 1.1: R Script for creating consensus genotypes

```
####----CREATING A GENOTYPE TABLE

#stringent conditions = three homozygotes are needed, two heterozygotes and one homozygote are accepted as
heterozygote
#data output = genotypes_table clean and all -> clean = only samples with less than 2 NA values (not more
than one locus missing)
#all = all sampleIDs that have been genotyped, also the ones that are completely missing (only useable for
NA analysis and not for allelematch)
#final / end version

library(dplyr)
library(tidyr)
library(naniar)
library(ggplot2)
library(data.table)
library(qwraps2)
library(arsenal)

#--------------------2014--------

data_2014_raw <- read.table("./Data/data_2014.txt", header=T, na.strings = "NA")
data_2014 <-
data.frame(data_2014_raw$SampleID,data_2014_raw$Marker,data_2014_raw$Allele1,data_2014_raw$Allele2)
#keep only coloumns that are needed
colnames(data_2014) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

str(data_2014)

#NA if data is ok
plot(data_2014$SampleID)

length(unique(data_2014$SampleID))
#[1] 316 -> there are 316 samples (faeces in the data of 2014)

nrow(data_2014)/length(unique(data_2014$SampleID))

data_2014_NA <- gather(data_2014, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_2014_N <- data_2014_NA[,c(1,2,3,6,4,7,5,8)]


#make distribution even -> add row with NA values so that every marker is mentioned exactly 7 times ->
works with integers but not with characters
data_2014_2 <- data_2014_N

data_2014_2$Marker <- as.integer(data_2014_N$Marker) #changes marker character string to integer numbers ->
needs to be changed back afterwards

DT=as.data.table(data_2014_2)
setkey(DT, SampleID, Marker)

data_2014_2 <- DT[CJ(unique(SampleID), seq(1:7))]

data_2014_2$Marker[data_2014_2$Marker == 1] <- "Lsa1"
data_2014_2$Marker[data_2014_2$Marker == 2] <- "Lsa3"
data_2014_2$Marker[data_2014_2$Marker == 3] <- "Sat5"
data_2014_2$Marker[data_2014_2$Marker == 4] <- "Sat8"
data_2014_2$Marker[data_2014_2$Marker == 5] <- "Sol30"
data_2014_2$Marker[data_2014_2$Marker == 6] <- "Sol33"
data_2014_2$Marker[data_2014_2$Marker == 7] <- "Sol8"

data_2014_N <- data_2014_2
summary(data_2014_N)
data_2014_N$Marker <- as.factor(data_2014_N$Marker)

plot(data_2014_N$Marker)

write.table(data_2014_N, file = "./Data/replicates data 2014.txt", sep = "\t", row.names = TRUE, col.names
= NA, quote=F)


data_2014_N[is.na(data_2014_N)] <- "NA"

Allele2_N_2014 <- ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA", data_2014_N$Allele2,
                  ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele2.1,

                  ifelse ((data_2014_N$Allele1!=data_2014_N$Allele1.1 |
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 | data_2014_N$Allele1.2!=data_2014_N$Allele1) &
(data_2014_N$Allele2!=data_2014_N$Allele2.1 | data_2014_N$Allele2.1!=data_2014_N$Allele2.2 &
data_2014_N$Allele2!=data_2014_N$Allele2.2) & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", "NA",
```

```
                ifelse ((data_2014_N$Allele1!=data_2014_N$Allele1.1 &
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 & data_2014_N$Allele1.2!=data_2014_N$Allele1) &
(data_2014_N$Allele2!=data_2014_N$Allele2.1 | data_2014_N$Allele2.1!=data_2014_N$Allele2.2 &
data_2014_N$Allele2!=data_2014_N$Allele2.2) & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", "NA",


                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2.2!=data_2014_N$Allele2 & data_2014_N$Allele2.1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2!=data_2014_N$Allele2.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1==data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.2 & data_2014_N$Allele2.1!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele2.2,

                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1.2!=data_2014_N$Allele1 & data_2014_N$Allele1.1==data_2014_N$Allele1 &
data_2014_N$Allele2.1==data_2014_N$Allele2 & data_2014_N$Allele2!="NA" & data_2014_N$Allele1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.2!="NA" &
data_2014_N$Allele1.2!="NA", data_2014_N$Allele2,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele2.1==data_2014_N$Allele2.2 & data_2014_N$Allele2!="NA" & data_2014_N$Allele1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.2!="NA" &
data_2014_N$Allele1.2!="NA", data_2014_N$Allele2.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele1.1!=data_2014_N$Allele1.2 &
data_2014_N$Allele2!="NA" & data_2014_N$Allele1!="NA" & data_2014_N$Allele2.1!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele2.2,

                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele2.2,
                ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele2.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele2,

                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.2,
                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.2 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.2,
                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1!=data_2014_N$Allele2.1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.1,
                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.1,
                # ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele2==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2,
                # ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele2==data_2014_N$Allele2.1 &
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele2,

                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele2==data_2014_N$Allele2.1 &
```

```
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.2=="NA" & data_2014_N$Allele2.2=="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA", data_2014_N$Allele2,
                ifelse (data_2014_N$Allele1.1!=data_2014_N$Allele2.1 &
data_2014_N$Allele1.2!=data_2014_N$Allele2.2 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1==data_2014_N$Allele1.2 & data_2014_N$Allele1=="NA" & data_2014_N$Allele2=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.2!=data_2014_N$Allele2.2 & data_2014_N$Allele2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele1.1=="NA" & data_2014_N$Allele2.1=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.2,
                ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele2==data_2014_N$Allele2.1 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.2=="NA" & data_2014_N$Allele2.2=="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA", data_2014_N$Allele2,
                ifelse (data_2014_N$Allele1.1==data_2014_N$Allele2.1 &
data_2014_N$Allele1.2==data_2014_N$Allele2.2 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1==data_2014_N$Allele1.2 & data_2014_N$Allele1=="NA" & data_2014_N$Allele2=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.1,
                ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.2==data_2014_N$Allele2.2 & data_2014_N$Allele2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele1.1=="NA" & data_2014_N$Allele2.1=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele2.2,"NA"))))))))))))))))))


Allele1_N_2014 <- ifelse(data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2&
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1,
                ifelse(data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1,

                ifelse ((data_2014_N$Allele1!=data_2014_N$Allele1.1 |
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 | data_2014_N$Allele1.2!=data_2014_N$Allele1) &
(data_2014_N$Allele2!=data_2014_N$Allele2.1 | data_2014_N$Allele2.1!=data_2014_N$Allele2.2 &
data_2014_N$Allele2!=data_2014_N$Allele2.2) & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", "NA",
                ifelse ((data_2014_N$Allele1!=data_2014_N$Allele1.1 &
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 & data_2014_N$Allele1.2!=data_2014_N$Allele1) &
(data_2014_N$Allele2!=data_2014_N$Allele2.1 | data_2014_N$Allele2.1!=data_2014_N$Allele2.2 &
data_2014_N$Allele2!=data_2014_N$Allele2.2) & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", "NA",


                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2.2!=data_2014_N$Allele2 & data_2014_N$Allele2.1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2!=data_2014_N$Allele2.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1==data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.2 & data_2014_N$Allele2.1!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1.2,

                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1.2!=data_2014_N$Allele1 & data_2014_N$Allele1.1==data_2014_N$Allele1 &
data_2014_N$Allele2.1==data_2014_N$Allele2 & data_2014_N$Allele2!="NA" & data_2014_N$Allele1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.2!="NA" &
data_2014_N$Allele1.2!="NA", data_2014_N$Allele1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele2.1==data_2014_N$Allele2.2 & data_2014_N$Allele2!="NA" & data_2014_N$Allele1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.2!="NA" &
data_2014_N$Allele1.2!="NA", data_2014_N$Allele1.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele1.1!=data_2014_N$Allele1.2 &
data_2014_N$Allele2!="NA" & data_2014_N$Allele1!="NA" & data_2014_N$Allele2.1!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1.2,

                ifelse(data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1.2,
                ifelse(data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1.1,
```

```
                ifelse(data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" & data_2014_N$Allele1.1!="NA" &
data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA",
data_2014_N$Allele1,

                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.2,
                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2!=data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.2 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.2,
                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1!=data_2014_N$Allele2.1 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.1,
                # ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele2==data_2014_N$Allele2.1 & data_2014_N$Allele1.1==data_2014_N$Allele1.2 &
data_2014_N$Allele1!=data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.1,
                # ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele2==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1,
                # ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele1.2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele2==data_2014_N$Allele2.1 &
data_2014_N$Allele1.1!=data_2014_N$Allele1.2 & data_2014_N$Allele1!="NA" & data_2014_N$Allele2!="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA" & data_2014_N$Allele1.2!="NA" &
data_2014_N$Allele2.2!="NA", data_2014_N$Allele1,

                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.1!=data_2014_N$Allele2.1 & data_2014_N$Allele2==data_2014_N$Allele2.1 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.2=="NA" & data_2014_N$Allele2.2=="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA", data_2014_N$Allele1,
                ifelse (data_2014_N$Allele1.1!=data_2014_N$Allele2.1 &
data_2014_N$Allele1.2!=data_2014_N$Allele2.2 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1==data_2014_N$Allele1.2 & data_2014_N$Allele1=="NA" & data_2014_N$Allele2=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.1,
                ifelse (data_2014_N$Allele1!=data_2014_N$Allele2 &
data_2014_N$Allele1.2!=data_2014_N$Allele2.2 & data_2014_N$Allele2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele1.1=="NA" & data_2014_N$Allele2.1=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.2,
                ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.1==data_2014_N$Allele2.1 & data_2014_N$Allele2==data_2014_N$Allele2.1 &
data_2014_N$Allele1==data_2014_N$Allele1.1 & data_2014_N$Allele1.2=="NA" & data_2014_N$Allele2.2=="NA" &
data_2014_N$Allele1.1!="NA" & data_2014_N$Allele2.1!="NA", data_2014_N$Allele1,
                ifelse (data_2014_N$Allele1.1==data_2014_N$Allele2.1 &
data_2014_N$Allele1.2==data_2014_N$Allele2.2 & data_2014_N$Allele2.1==data_2014_N$Allele2.2 &
data_2014_N$Allele1.1==data_2014_N$Allele1.2 & data_2014_N$Allele1=="NA" & data_2014_N$Allele2=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.1,
                ifelse (data_2014_N$Allele1==data_2014_N$Allele2 &
data_2014_N$Allele1.2==data_2014_N$Allele2.2 & data_2014_N$Allele2==data_2014_N$Allele2.2 &
data_2014_N$Allele1==data_2014_N$Allele1.2 & data_2014_N$Allele1.1=="NA" & data_2014_N$Allele2.1=="NA" &
data_2014_N$Allele1.2!="NA" & data_2014_N$Allele2.2!="NA", data_2014_N$Allele1.2,"NA"))))))))))))))))))

gt_N_14 <- cbind(data_2014_N,Allele1_N_2014,Allele2_N_2014)

gt_N_14[ gt_N_14 == "NA" ] <- NA

gt.wrep_14 <- write.table(gt_N_14, file = "./Data/gt.wreps_14.txt", sep = "\t", row.names = F, col.names =
T, quote=F)

new_DF_2014 <- gt_N_14[rowSums(is.na(gt_N_14)) > 0,] #shows where the missing values are

gt.tab_14.n <- gt_N_14[,c(1,2,9,10)]

# vis_dat(gt_tab_14.n)
#
# gg_miss_upset(gt_tab_14.n) #shows the distributions and combinations of missing data in the data frame
# gg_miss_var(gt_tab_14.n,
#             facet = Marker) #shows the missing values for the different markers -> Lsa1 has the most
missing values

gt.tab_14.n$Allele1_N_2014 <- as.numeric(as.character(gt.tab_14.n$Allele1_N_2014))
gt.tab_14.n$Allele2_N_2014 <- as.numeric(as.character(gt.tab_14.n$Allele2_N_2014))

plot(gt.tab_14.n$Allele2_N_2014~gt.tab_14.n$Allele1_N_2014)

gt.raw_14 <- gather(gt.tab_14.n, key = "k", value = "v", -SampleID, na.rm = F) %>%
  group_by(SampleID) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

gt.input_14 <- gt.raw_14[,c(1,2,9,3,10,4,11,5,12,6,13,7,14,8,15)]
colnames(gt.input_14) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2")       #rename coloumns

sex_14 <- read.table("./Data/results.sexdet_14.txt", header=T, na.strings = "NA")
```

```
compare(sex_14$SampleID, gt.input_14$SampleID)     #to make sure that they are the same and that the sex is
really for the same sample

gt.result_14 <- cbind(gt.input_14,sex_14$sex.14)

gt.clean_14 <- gt.result_14[-which(rowMeans(is.na(gt.result_14)) > 0.2), ]
#gt_clean.14$na_count <- apply(is.na(gt_clean.14), 1, sum)

colnames(gt.result_14) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2", "sex")       #rename
coloumns


write.table(gt.clean_14, file = "./Data/genotypestable.clean_2014.txt", sep = "\t", row.names = F,
col.names = T, quote=F)
write.table(gt.result_14, file = "./Data/genotypestable.all_2014.txt", sep = "\t", row.names = F, col.names
= T, quote=F)



#--------------------2015--------

data_2015_raw <- read.table("./Data/data_2015.txt", header=T, na.strings = "NA")
data_2015 <-
data.frame(data_2015_raw$SampleID,data_2015_raw$Marker,data_2015_raw$Allele1,data_2015_raw$Allele2)
#keep only coloumns that are needed
colnames(data_2015) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

#NA if data is ok
plot(data_2015$SampleID)


length(unique(data_2015$SampleID))
#[1] 254 -> there are 254 samples (faeces in the data of 2015)

nrow(data_2015)/length(unique(data_2015$SampleID))
#[1] 21 -> every sample is mentioned 21 times -> o.k.


#rearrange data so that all alleles are in one row
data_2015_NA <- gather(data_2015, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_2015_N <- data_2015_NA[,c(1,2,3,6,4,7,5,8)]

#make distribution even -> add row with NA values so that every marker is mentioned exactly 7 times ->
works with integers but not with characters
data_2015_2 <- data_2015_N
data_2015_2$Marker <- as.integer(data_2015_N$Marker) #changes marker character string to integer numbers ->
needs to be changed back afterwards

DT=as.data.table(data_2015_2)
setkey(DT, SampleID, Marker)

data_2015_2 <- DT[CJ(unique(SampleID), seq(1:7))]

data_2015_2$Marker[data_2015_2$Marker == 1] <- "Lsa1"
data_2015_2$Marker[data_2015_2$Marker == 2] <- "Lsa3"
data_2015_2$Marker[data_2015_2$Marker == 3] <- "Sat5"
data_2015_2$Marker[data_2015_2$Marker == 4] <- "Sat8"
data_2015_2$Marker[data_2015_2$Marker == 5] <- "Sol30"
data_2015_2$Marker[data_2015_2$Marker == 6] <- "Sol33"
data_2015_2$Marker[data_2015_2$Marker == 7] <- "Sol8"

data_2015_N <- data_2015_2
summary(data_2015_N)
data_2015_N$Marker <- as.factor(data_2015_N$Marker)

plot(data_2015_N$Marker)           #every marker has same frequency
summary(data_2015_N)                #good -> every sampleID has frequency of 7, marker 254 -> 254 unique
sampleIDs -> one marker for every sample -> good

write.table(data_2015_N, file = "replicates data 2015", sep = "\t", row.names = TRUE, col.names = NA,
quote=F)

# vis_dat(data_2015_N)
#
# gg_miss_upset(data_2015_N) #shows the distributions and combinations of missing data in the data frame ->
the 4th replication is missing in 280 markers, the there are 172 times only two replicates per sample and
marker
# gg_miss_var(data_2015_N,
#             facet = Marker) #shows the missing values for the different markers -> Sol33 has the most
missing values
#
# ggplot(data_2015_N,
#        aes(x = Allele1.2,
#            y = Allele1.1)) +
#   geom_miss_point() +
#   facet_wrap(~Marker)
#


#change NA to "NA" -> not missing value but character
# data_2015_N$Allele1 <- as.character(data_2015_N$Allele1)
# data_2015_N$Allele2 <- as.character(data_2015_N$Allele2)
# data_2015_N$Allele1.1 <- as.character(data_2015_N$Allele1.1)
# data_2015_N$Allele2.1 <- as.character(data_2015_N$Allele2.1)
# data_2015_N$Allele1.2 <- as.character(data_2015_N$Allele1.2)
# data_2015_N$Allele2.2 <- as.character(data_2015_N$Allele2.2)
```

```
data_2015_N[is.na(data_2015_N)] <- "NA"

Allele2_N_2015 <- ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA", data_2015_N$Allele2,
                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele2.1,

                # ifelse ((data_2015_N$Allele1!=data_2015_N$Allele1.1 |
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 | data_2015_N$Allele1.2!=data_2015_N$Allele1) &
(data_2015_N$Allele2!=data_2015_N$Allele2.1 | data_2015_N$Allele2.1!=data_2015_N$Allele2.2 &
data_2015_N$Allele2!=data_2015_N$Allele2.2) & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", "NA",
                  ifelse ((data_2015_N$Allele1!=data_2015_N$Allele1.1 &
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 & data_2015_N$Allele1.2!=data_2015_N$Allele1) &
(data_2015_N$Allele2!=data_2015_N$Allele2.1 | data_2015_N$Allele2.1!=data_2015_N$Allele2.2 &
data_2015_N$Allele2!=data_2015_N$Allele2.2) & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", "NA",

                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2.2!=data_2015_N$Allele2 & data_2015_N$Allele2.1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2,
                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2!=data_2015_N$Allele2.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1==data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.1,
                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.2 & data_2015_N$Allele2.1!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele2.2,

                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1.2!=data_2015_N$Allele1 & data_2015_N$Allele1.1==data_2015_N$Allele1 &
data_2015_N$Allele2.1==data_2015_N$Allele2 & data_2015_N$Allele2!="NA" & data_2015_N$Allele1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.2!="NA" &
data_2015_N$Allele1.2!="NA", data_2015_N$Allele2,
                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele2.1==data_2015_N$Allele2.2 & data_2015_N$Allele2!="NA" & data_2015_N$Allele1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.2!="NA" &
data_2015_N$Allele1.2!="NA", data_2015_N$Allele2.1,
                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele1.1!=data_2015_N$Allele1.2 &
data_2015_N$Allele2!="NA" & data_2015_N$Allele1!="NA" & data_2015_N$Allele2.1!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele2.2,

                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele2.2,
                  ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele2.1,
                  ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele2,

                  # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.2,
                  # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.2 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.2,
                  # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1!=data_2015_N$Allele2.1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.1,
```

```
                # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.1,
                # ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele2==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2,
                # ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele2==data_2015_N$Allele2.1 &
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele2,

                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele2==data_2015_N$Allele2.1 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.2=="NA" & data_2015_N$Allele2.2=="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA", data_2015_N$Allele2,
                ifelse (data_2015_N$Allele1.1!=data_2015_N$Allele2.1 &
data_2015_N$Allele1.2!=data_2015_N$Allele2.2 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1==data_2015_N$Allele1.2 & data_2015_N$Allele1=="NA" & data_2015_N$Allele2=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.1,
                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.2!=data_2015_N$Allele2.2 & data_2015_N$Allele2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele1.1=="NA" & data_2015_N$Allele2.1=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.2,
                ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele2==data_2015_N$Allele2.1 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.2=="NA" & data_2015_N$Allele2.2=="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA", data_2015_N$Allele2,
                ifelse (data_2015_N$Allele1.1==data_2015_N$Allele2.1 &
data_2015_N$Allele1.2==data_2015_N$Allele2.2 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1==data_2015_N$Allele1.2 & data_2015_N$Allele1=="NA" & data_2015_N$Allele2=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.1,
                ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.2==data_2015_N$Allele2.2 & data_2015_N$Allele2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele1.1=="NA" & data_2015_N$Allele2.1=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele2.2,"NA"))))))))))))))))))

Allele1_N_2015 <- ifelse(data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2&
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1,
                ifelse(data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1,

               # ifelse ((data_2015_N$Allele1!=data_2015_N$Allele1.1 |
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 | data_2015_N$Allele1.2!=data_2015_N$Allele1) &
(data_2015_N$Allele2!=data_2015_N$Allele2.1 | data_2015_N$Allele2.1!=data_2015_N$Allele2.2 &
data_2015_N$Allele2!=data_2015_N$Allele2.2) & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", "NA",
                ifelse ((data_2015_N$Allele1!=data_2015_N$Allele1.1 &
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 & data_2015_N$Allele1.2!=data_2015_N$Allele1) &
(data_2015_N$Allele2!=data_2015_N$Allele2.1 | data_2015_N$Allele2.1!=data_2015_N$Allele2.2 &
data_2015_N$Allele2!=data_2015_N$Allele2.2) & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", "NA",

                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2.2!=data_2015_N$Allele2 & data_2015_N$Allele2.1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1,
                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2!=data_2015_N$Allele2.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1==data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.1,
                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.2 & data_2015_N$Allele2.1!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1.2,

                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1.2!=data_2015_N$Allele1 & data_2015_N$Allele1.1==data_2015_N$Allele1 &
data_2015_N$Allele2.1==data_2015_N$Allele2 & data_2015_N$Allele2!="NA" & data_2015_N$Allele1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.2!="NA" &
data_2015_N$Allele1.2!="NA", data_2015_N$Allele1,
                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele2.1==data_2015_N$Allele2.2 & data_2015_N$Allele2!="NA" & data_2015_N$Allele1!="NA" &
```

```
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.2!="NA" &
data_2015_N$Allele1.2!="NA", data_2015_N$Allele1.1,
                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele1.1!=data_2015_N$Allele1.2 &
data_2015_N$Allele2!="NA" & data_2015_N$Allele1!="NA" & data_2015_N$Allele2.1!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1.2,

                ifelse(data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1.2,
                ifelse(data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1.1,
                ifelse(data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" & data_2015_N$Allele1.1!="NA" &
data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA",
data_2015_N$Allele1,

                # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.2,
                # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2!=data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.2 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.2,
                # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1!=data_2015_N$Allele2.1 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.1,
                # ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele2==data_2015_N$Allele2.1 & data_2015_N$Allele1.1==data_2015_N$Allele1.2 &
data_2015_N$Allele1!=data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.1,
                # ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele2==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1,
                # ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele1.2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele2==data_2015_N$Allele2.1 &
data_2015_N$Allele1.1!=data_2015_N$Allele1.2 & data_2015_N$Allele1!="NA" & data_2015_N$Allele2!="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA" & data_2015_N$Allele1.2!="NA" &
data_2015_N$Allele2.2!="NA", data_2015_N$Allele1,

                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.1!=data_2015_N$Allele2.1 & data_2015_N$Allele2==data_2015_N$Allele2.1 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.2=="NA" & data_2015_N$Allele2.2=="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA", data_2015_N$Allele1,
                ifelse (data_2015_N$Allele1.1!=data_2015_N$Allele2.1 &
data_2015_N$Allele1.2!=data_2015_N$Allele2.2 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1==data_2015_N$Allele1.2 & data_2015_N$Allele1=="NA" & data_2015_N$Allele2=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.1,
                ifelse (data_2015_N$Allele1!=data_2015_N$Allele2 &
data_2015_N$Allele1.2!=data_2015_N$Allele2.2 & data_2015_N$Allele2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele1.1=="NA" & data_2015_N$Allele2.1=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.2,
                ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.1==data_2015_N$Allele2.1 & data_2015_N$Allele2==data_2015_N$Allele2.1 &
data_2015_N$Allele1==data_2015_N$Allele1.1 & data_2015_N$Allele1.2=="NA" & data_2015_N$Allele2.2=="NA" &
data_2015_N$Allele1.1!="NA" & data_2015_N$Allele2.1!="NA", data_2015_N$Allele1,
                ifelse (data_2015_N$Allele1.1==data_2015_N$Allele2.1 &
data_2015_N$Allele1.2==data_2015_N$Allele2.2 & data_2015_N$Allele2.1==data_2015_N$Allele2.2 &
data_2015_N$Allele1.1==data_2015_N$Allele1.2 & data_2015_N$Allele1=="NA" & data_2015_N$Allele2=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.1,
                ifelse (data_2015_N$Allele1==data_2015_N$Allele2 &
data_2015_N$Allele1.2==data_2015_N$Allele2.2 & data_2015_N$Allele2==data_2015_N$Allele2.2 &
data_2015_N$Allele1==data_2015_N$Allele1.2 & data_2015_N$Allele1.1=="NA" & data_2015_N$Allele2.1=="NA" &
data_2015_N$Allele1.2!="NA" & data_2015_N$Allele2.2!="NA", data_2015_N$Allele1.2,"NA"))))))))))))))))))


gt_N_15 <- cbind(data_2015_N,Allele1_N_2015,Allele2_N_2015)
gt_N_15[ gt_N_15 == "NA" ] <- NA

new_DF_2015 <- gt_N_15[rowSums(is.na(gt_N_15)) > 0,] #shows where the missing values are

gt.tab_15.n <- gt_N_15[,c(1,2,9,10)]

gt.wrep_15 <- write.table(gt_N_15, file = "./Data/gt.wreps_15.txt", sep = "\t", row.names = F, col.names =
T, quote=F)


summary(gt_N_15)
# vis_dat(gt_tab_15.n)
```

```
#
# gg_miss_upset(gt_tab_15.n) #shows the distributions and combinations of missing data in the data frame
# gg_miss_var(gt_tab_15.n,
#             facet = Marker) #shows the missing values for the different markers -> Lsa1 has the most
missing values

gt.tab_15.n$Allele1_N_2015 <- as.numeric(as.character(gt.tab_15.n$Allele1_N_2015))
gt.tab_15.n$Allele2_N_2015 <- as.numeric(as.character(gt.tab_15.n$Allele2_N_2015))

plot(gt.tab_15.n$Allele2_N_2015~gt.tab_15.n$Allele1_N_2015)

gt.raw_15 <- gather(gt.tab_15.n, key = "k", value = "v", -SampleID, na.rm = F) %>%
  group_by(SampleID) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

gt.input_15 <- gt.raw_15[,c(1,2,9,3,10,4,11,5,12,6,13,7,14,8,15)]
colnames(gt.input_15) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2")       #rename coloumns

sex_15 <- read.table("./Data/results.sexdet_15.txt", header=T, na.strings = "NA")
compare(sex_15$SampleID, gt.input_15$SampleID)     #to make sure that they are the same and that the sex is
really for the same sample

gt.result_15 <- cbind(gt.input_15,sex_15$sex_15)

gt.clean_15 <- gt.result_15[-which(rowMeans(is.na(gt.result_15)) > 0.2), ]
#gt_clean.15$na_count <- apply(is.na(gt_clean.15), 1, sum)

colnames(gt.result_15) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2", "sex")       #rename
coloumns

write.table(gt.clean_15, file = "./Data/genotypestable.clean_2015.txt", sep = "\t", row.names = F,
col.names = T, quote=F)
write.table(gt.result_15, file = "./Data/genotypestable.all_2015.txt", sep = "\t", row.names = F, col.names
= T, quote=F)



#--------------------2016--------------------

data_2016_raw <- read.table("./Data/data_2016.txt", header=T, na.strings = "NA")
data_2016 <-
data.frame(data_2016_raw$SampleID,data_2016_raw$Marker,data_2016_raw$Allele1,data_2016_raw$Allele2)
#keep only coloumns that are needed
colnames(data_2016) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

#NA if data is ok
plot(data_2016$SampleID)

length(unique(data_2016$SampleID))
#[1] 336 -> there are 336 samples (faeces in the data of 2016)

nrow(data_2016)/length(unique(data_2016$SampleID))
#[1] 20.73 -> uneven distribution


#rearrange data so that all alleles are in one row
data_2016_NA <- gather(data_2016, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_2016_N <- data_2016_NA[,c(1,2,3,6,4,7,5,8)]


#make distribution even -> add row with NA values so that every marker is mentioned exactly 7 times ->
works with integers but not with characters
data_2016_2 <- data_2016_N
data_2016_2$Marker <- as.integer(data_2016_N$Marker) #changes marker character string to integer numbers ->
needs to be changed back afterwards

DT=as.data.table(data_2016_2)
setkey(DT, SampleID, Marker)

data_2016_2 <- DT[CJ(unique(SampleID), seq(1:7))]

data_2016_2$Marker[data_2016_2$Marker == 1] <- "Lsa1"
data_2016_2$Marker[data_2016_2$Marker == 2] <- "Lsa3"
data_2016_2$Marker[data_2016_2$Marker == 3] <- "Sat5"
data_2016_2$Marker[data_2016_2$Marker == 4] <- "Sat8"
data_2016_2$Marker[data_2016_2$Marker == 5] <- "Sol30"
data_2016_2$Marker[data_2016_2$Marker == 6] <- "Sol33"
data_2016_2$Marker[data_2016_2$Marker == 7] <- "Sol8"

data_2016_N <- data_2016_2
summary(data_2016_N)
data_2016_N$Marker <- as.factor(data_2016_N$Marker)

write.table(data_2016_N, file = "replicates data 2016.txt", sep = "\t", row.names = F, col.names = T,
quote=F)


# vis_dat(data_2016_N)
#
# gg_miss_upset(data_2016_N) #shows the distributions and combinations of missing data in the data frame ->
the 4th replication is missing in 280 markers, the there are 172 times only two replicates per sample and
marker
# gg_miss_var(data_2016_N,
```

```
#             facet = Marker) #shows the missing values for the different markers -> Sol30 has the most
missing values
#
# ggplot(data_2016_N,
#        aes(x = Allele1.2,
#            y = Allele1.1)) +
#   geom_miss_point() +
#   facet_wrap(~Marker)


#change NA to "NA" -> not missing value but characters

# data_2016_N$Allele1 <- as.numeric(as.character(data_2016_N$Allele1))
# data_2016_N$Allele2 <- as.numeric(as.character(data_2016_N$Allele2))
# data_2016_N$Allele1.1 <- as.numeric(as.character(data_2016_N$Allele1.1))
# data_2016_N$Allele2.1 <- as.numeric(as.character(data_2016_N$Allele2.1))
# data_2016_N$Allele1.2 <- as.numeric(as.character(data_2016_N$Allele1.2))
# data_2016_N$Allele2.2 <- as.numeric(as.character(data_2016_N$Allele2.2))

data_2016_N[is.na(data_2016_N)] <- "NA"

Allele2_N_2016 <- ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA", data_2016_N$Allele2,
                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele2.1,

                 #ifelse ((data_2016_N$Allele1!=data_2016_N$Allele1.1 |
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 | data_2016_N$Allele1.2!=data_2016_N$Allele1) &
(data_2016_N$Allele2!=data_2016_N$Allele2.1 | data_2016_N$Allele2.1!=data_2016_N$Allele2.2 &
data_2016_N$Allele2!=data_2016_N$Allele2.2) & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", "NA",
                  ifelse ((data_2016_N$Allele1!=data_2016_N$Allele1.1 &
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 & data_2016_N$Allele1.2!=data_2016_N$Allele1) &
(data_2016_N$Allele2!=data_2016_N$Allele2.1 | data_2016_N$Allele2.1!=data_2016_N$Allele2.2 &
data_2016_N$Allele2!=data_2016_N$Allele2.2) & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", "NA",


                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2.2!=data_2016_N$Allele2 & data_2016_N$Allele2.1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2,
                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2!=data_2016_N$Allele2.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1==data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.1,
                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.2 & data_2016_N$Allele2.1!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele2.2,

                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1.2!=data_2016_N$Allele1 & data_2016_N$Allele1.1==data_2016_N$Allele1 &
data_2016_N$Allele2.1==data_2016_N$Allele2 & data_2016_N$Allele2!="NA" & data_2016_N$Allele1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.2!="NA" &
data_2016_N$Allele1.2!="NA", data_2016_N$Allele2,
                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele2.1==data_2016_N$Allele2.2 & data_2016_N$Allele2!="NA" & data_2016_N$Allele1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.2!="NA" &
data_2016_N$Allele1.2!="NA", data_2016_N$Allele2.1,
                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele1.1!=data_2016_N$Allele1.2 &
data_2016_N$Allele2!="NA" & data_2016_N$Allele1!="NA" & data_2016_N$Allele2.1!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele2.2,

                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele2.2,
                  ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele2.1,
                  ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele2,
```

```
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.2,
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.2 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.2,
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele2.1 & data_2016_N$Allele1.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1!=data_2016_N$Allele2.1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.1,
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.1,
                # ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele2==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2,
                # ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele2==data_2016_N$Allele2.1 &
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele2,

                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele2==data_2016_N$Allele2.1 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.2=="NA" & data_2016_N$Allele2.2=="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA", data_2016_N$Allele2,
                ifelse (data_2016_N$Allele1.1!=data_2016_N$Allele2.1 &
data_2016_N$Allele1.2!=data_2016_N$Allele2.2 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1==data_2016_N$Allele1.2 & data_2016_N$Allele1=="NA" & data_2016_N$Allele2=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.1,
                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.2!=data_2016_N$Allele2.2 & data_2016_N$Allele2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele1.1=="NA" & data_2016_N$Allele2.1=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.2,
                ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele2==data_2016_N$Allele2.1 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.2=="NA" & data_2016_N$Allele2.2=="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA", data_2016_N$Allele2,
                ifelse (data_2016_N$Allele1.1==data_2016_N$Allele2.1 &
data_2016_N$Allele1.2==data_2016_N$Allele2.2 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1==data_2016_N$Allele1.2 & data_2016_N$Allele1=="NA" & data_2016_N$Allele2=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.1,
                ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.2==data_2016_N$Allele2.2 & data_2016_N$Allele2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele1.1=="NA" & data_2016_N$Allele2.1=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele2.2,"NA")))))))))))))))))

Allele1_N_2016 <- ifelse(data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2&
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1,
                ifelse(data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1,

               #ifelse ((data_2016_N$Allele1!=data_2016_N$Allele1.1 |
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 | data_2016_N$Allele1.2!=data_2016_N$Allele1) &
(data_2016_N$Allele2!=data_2016_N$Allele2.1 | data_2016_N$Allele2.1!=data_2016_N$Allele2.2 &
data_2016_N$Allele2!=data_2016_N$Allele2.2) & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", "NA",
                ifelse ((data_2016_N$Allele1!=data_2016_N$Allele1.1 &
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 & data_2016_N$Allele1.2!=data_2016_N$Allele1) &
(data_2016_N$Allele2!=data_2016_N$Allele2.1 | data_2016_N$Allele2.1!=data_2016_N$Allele2.2 &
data_2016_N$Allele2!=data_2016_N$Allele2.2) & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", "NA",

                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2.2!=data_2016_N$Allele2 & data_2016_N$Allele2.1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1,
                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2!=data_2016_N$Allele2.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1==data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.1,
```

```
                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.2 & data_2016_N$Allele2.1!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1.2,

                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1.2!=data_2016_N$Allele1 & data_2016_N$Allele1.1==data_2016_N$Allele1 &
data_2016_N$Allele2.1==data_2016_N$Allele2 & data_2016_N$Allele2!="NA" & data_2016_N$Allele1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.2!="NA" &
data_2016_N$Allele1.2!="NA", data_2016_N$Allele1,
                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele2.1==data_2016_N$Allele2.2 & data_2016_N$Allele2!="NA" & data_2016_N$Allele1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.2!="NA" &
data_2016_N$Allele1.2!="NA", data_2016_N$Allele1.1,
                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele1.1!=data_2016_N$Allele1.2 &
data_2016_N$Allele2!="NA" & data_2016_N$Allele1!="NA" & data_2016_N$Allele2.1!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1.2,

                ifelse(data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1.2,
                ifelse(data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1.1,
                ifelse(data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" & data_2016_N$Allele1.1!="NA" &
data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA",
data_2016_N$Allele1,

                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.2,
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2!=data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.2 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.2,
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1!=data_2016_N$Allele2.1 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.1,
                # ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele2==data_2016_N$Allele2.1 & data_2016_N$Allele1.1==data_2016_N$Allele1.2 &
data_2016_N$Allele1!=data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.1,
                # ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele2==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1,
                # ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele1.2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele2==data_2016_N$Allele2.1 &
data_2016_N$Allele1.1!=data_2016_N$Allele1.2 & data_2016_N$Allele1!="NA" & data_2016_N$Allele2!="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA" & data_2016_N$Allele1.2!="NA" &
data_2016_N$Allele2.2!="NA", data_2016_N$Allele1,

                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.1!=data_2016_N$Allele2.1 & data_2016_N$Allele2==data_2016_N$Allele2.1 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.2=="NA" & data_2016_N$Allele2.2=="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA", data_2016_N$Allele1,
                ifelse (data_2016_N$Allele1.1!=data_2016_N$Allele2.1 &
data_2016_N$Allele1.2!=data_2016_N$Allele2.2 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1==data_2016_N$Allele1.2 & data_2016_N$Allele1=="NA" & data_2016_N$Allele2=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.1,
                ifelse (data_2016_N$Allele1!=data_2016_N$Allele2 &
data_2016_N$Allele1.2!=data_2016_N$Allele2.2 & data_2016_N$Allele2==data_2016_N$Allele2.2 &
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele1.1=="NA" & data_2016_N$Allele2.1=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.2,
                ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.1==data_2016_N$Allele2.1 & data_2016_N$Allele2==data_2016_N$Allele2.1 &
data_2016_N$Allele1==data_2016_N$Allele1.1 & data_2016_N$Allele1.2=="NA" & data_2016_N$Allele2.2=="NA" &
data_2016_N$Allele1.1!="NA" & data_2016_N$Allele2.1!="NA", data_2016_N$Allele1,
                ifelse (data_2016_N$Allele1.1==data_2016_N$Allele2.1 &
data_2016_N$Allele1.2==data_2016_N$Allele2.2 & data_2016_N$Allele2.1==data_2016_N$Allele2.2 &
data_2016_N$Allele1.1==data_2016_N$Allele1.2 & data_2016_N$Allele1=="NA" & data_2016_N$Allele2=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.1,
                ifelse (data_2016_N$Allele1==data_2016_N$Allele2 &
data_2016_N$Allele1.2==data_2016_N$Allele2.2 & data_2016_N$Allele2==data_2016_N$Allele2.2 &
```

```
data_2016_N$Allele1==data_2016_N$Allele1.2 & data_2016_N$Allele1.1=="NA" & data_2016_N$Allele2.1=="NA" &
data_2016_N$Allele1.2!="NA" & data_2016_N$Allele2.2!="NA", data_2016_N$Allele1.2,"NA")))))))))))))))))


gt_N_16 <- cbind(data_2016_N,Allele1_N_2016,Allele2_N_2016)

gt_N_16[ gt_N_16 == "NA" ] <- NA

write.table(gt_N_16, file = "./Data/gt.wreps_16.txt", sep = "\t", row.names = F, col.names = T, quote=F)


new_DF_2016 <- gt_N_16[rowSums(is.na(gt_N_16)) > 0,] #shows where the missing values are

gt_tab_16.n <- gt_N_16[,c(1,2,9,10)]

# vis_dat(gt_tab_16.n)
#
# gg_miss_upset(gt_tab_16.n) #shows the distributions and combinations of missing data in the data frame
# gg_miss_var(gt_tab_16.n,
#             facet = Marker) #shows the missing values for the different markers -> Lsa1 has the most
missing values

gt_tab_16.n$Allele1_N_2016 <- as.numeric(as.character(gt_tab_16.n$Allele1_N_2016))
gt_tab_16.n$Allele2_N_2016 <- as.numeric(as.character(gt_tab_16.n$Allele2_N_2016))

plot(gt_tab_16.n$Allele2_N_2016~gt_tab_16.n$Allele1_N_2016)

gt.raw_16 <- gather(gt_tab_16.n, key = "k", value = "v", -SampleID, na.rm = F) %>%
  group_by(SampleID) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

gt.input_16 <- gt.raw_16[,c(1,2,9,3,10,4,11,5,12,6,13,7,14,8,15)]
colnames(gt.input_16) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.2", "Sat5.1",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2")       #rename coloumns

sex_16 <- read.table("./Data/results.sexdet_16.txt", header=T, na.strings = "NA")
compare(sex_16$SampleID, gt.input_16$SampleID)     #to make sure that they are the same and that the sex is
really for the same sample

gt.result_16 <- cbind(gt.input_16,sex_16$sex_16)

gt.clean_16 <- gt.result_16[-which(rowMeans(is.na(gt.result_16)) > 0.2), ]
#gt_clean.16$na_count <- apply(is.na(gt_clean.16), 1, sum)

colnames(gt.result_16) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2", "sex")       #rename
coloumns


write.table(gt.clean_16, file = "./Data/genotypestable.clean_2016.txt", sep = "\t", row.names = F,
col.names = T, quote=F)
write.table(gt.result_16, file = "./Data/genotypestable.all_2016.txt", sep = "\t", row.names = F, col.names
= T, quote=F)


#--------------------2017----------------------

data_2017_raw <- read.table("./Data/data_2017.txt", header=T, na.strings = "NA")
data_2017 <-
data.frame(data_2017_raw$SampleID,data_2017_raw$Marker,data_2017_raw$Allele1,data_2017_raw$Allele2)
#keep only coloumns that are needed
colnames(data_2017) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

#NA if data is ok
plot(data_2017$SampleID)

length(unique(data_2017$SampleID))
#[1] 345 -> there are 345 samples (faeces in the data of 2017)

nrow(data_2017)/length(unique(data_2017$SampleID))
#[1] 19.25507 -> not ok, unequal distribution of markers


#rearrange data so that all alleles are in one row
data_2017_NA <- gather(data_2017, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_2017_N <- data_2017_NA[,c(1,2,3,6,4,7,5,8)]
summary(data_2017_N)
plot(data_2017_N$Marker)

#make distribution even -> add row with NA values so that every marker is mentioned exactly 7 times ->
works with integers but not with characters
data_2017_2 <- data_2017_N
data_2017_2$Marker <- as.integer(data_2017_N$Marker) #changes marker character string to integer numbers ->
needs to be changed back afterwards

DT=as.data.table(data_2017_2)
setkey(DT, SampleID, Marker)

data_2017_2 <- DT[CJ(unique(SampleID), seq(1:7))]

data_2017_2$Marker[data_2017_2$Marker == 1] <- "Lsa1"
data_2017_2$Marker[data_2017_2$Marker == 2] <- "Lsa3"
data_2017_2$Marker[data_2017_2$Marker == 3] <- "Sat5"
data_2017_2$Marker[data_2017_2$Marker == 4] <- "Sat8"
```

```
data_2017_2$Marker[data_2017_2$Marker == 5] <- "Sol30"
data_2017_2$Marker[data_2017_2$Marker == 6] <- "Sol33"
data_2017_2$Marker[data_2017_2$Marker == 7] <- "Sol8"

data_2017_N <- data_2017_2
#summary(data_2017_N)
data_2017_N$Marker <- as.factor(data_2017_N$Marker)
summary(data_2017_N)

write.table(data_2017_N, file = "replicates data 2017.txt", sep = "\t", row.names = F, col.names = T,
quote=F)


# vis_dat(data_2017_N)
#
# gg_miss_upset(data_2017_N) #shows the distributions and combinations of missing data in the data frame ->
the 4th replication is missing in 280 markers, the there are 172 times only two replicates per sample and
marker
# gg_miss_var(data_2017_N,
#             facet = Marker) #shows the missing values for the different markers -> Sol30 has the most
missing values

# ggplot(data_2017_N,
#        aes(x = Allele1.2,
#            y = Allele1.1)) +
#   geom_miss_point() +
#   facet_wrap(~Marker)


#change NA to "NA" -> not missing value but characters

# data_2017_N$Allele1 <- as.numeric(as.character(data_2017_N$Allele1))
# data_2017_N$Allele2 <- as.numeric(as.character(data_2017_N$Allele2))
# data_2017_N$Allele1.1 <- as.numeric(as.character(data_2017_N$Allele1.1))
# data_2017_N$Allele2.1 <- as.numeric(as.character(data_2017_N$Allele2.1))
# data_2017_N$Allele1.2 <- as.numeric(as.character(data_2017_N$Allele1.2))
# data_2017_N$Allele2.2 <- as.numeric(as.character(data_2017_N$Allele2.2))

data_2017_N[is.na(data_2017_N)] <- "NA"

Allele2_N_2017 <- ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA", data_2017_N$Allele2,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.1,

                #ifelse ((data_2017_N$Allele1!=data_2017_N$Allele1.1 |
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 | data_2017_N$Allele1.2!=data_2017_N$Allele1) &
(data_2017_N$Allele2!=data_2017_N$Allele2.1 | data_2017_N$Allele2.1!=data_2017_N$Allele2.2 &
data_2017_N$Allele2!=data_2017_N$Allele2.2) & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", "NA",
                ifelse ((data_2017_N$Allele1!=data_2017_N$Allele1.1 &
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 & data_2017_N$Allele1.2!=data_2017_N$Allele1) &
(data_2017_N$Allele2!=data_2017_N$Allele2.1 | data_2017_N$Allele2.1!=data_2017_N$Allele2.2 &
data_2017_N$Allele2!=data_2017_N$Allele2.2) & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", "NA",

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2.2!=data_2017_N$Allele2 & data_2017_N$Allele2.1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2!=data_2017_N$Allele2.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1==data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.2 & data_2017_N$Allele2.1!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.2,

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1.2!=data_2017_N$Allele1 & data_2017_N$Allele1.1==data_2017_N$Allele1 &
data_2017_N$Allele2.1==data_2017_N$Allele2 & data_2017_N$Allele2!="NA" & data_2017_N$Allele1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.2!="NA" &
data_2017_N$Allele1.2!="NA", data_2017_N$Allele2,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele2.1==data_2017_N$Allele2.2 & data_2017_N$Allele2!="NA" & data_2017_N$Allele1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.2!="NA" &
data_2017_N$Allele1.2!="NA", data_2017_N$Allele2.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele1.1!=data_2017_N$Allele1.2 &
data_2017_N$Allele2!="NA" & data_2017_N$Allele1!="NA" & data_2017_N$Allele2.1!="NA" &
```

```
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.2,

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.2,
                ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2,

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.2,
                ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele2,

                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.2,
                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.2 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.2,
                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1!=data_2017_N$Allele2.1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.1,
                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.1,
                # ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele2==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2,
                # ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele2==data_2017_N$Allele2.1 &
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele2,

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele2==data_2017_N$Allele2.1 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.2==="NA" & data_2017_N$Allele2.2=="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA", data_2017_N$Allele2,
                ifelse (data_2017_N$Allele1.1!=data_2017_N$Allele2.1 &
data_2017_N$Allele1.2!=data_2017_N$Allele2.2 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1==data_2017_N$Allele1.2 & data_2017_N$Allele1=="NA" & data_2017_N$Allele2=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.2!=data_2017_N$Allele2.2 & data_2017_N$Allele2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele1.1=="NA" & data_2017_N$Allele2.1=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.2,
                ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele2==data_2017_N$Allele2.1 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.2==="NA" & data_2017_N$Allele2.2=="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA", data_2017_N$Allele2,
                ifelse (data_2017_N$Allele1.1==data_2017_N$Allele2.1 &
data_2017_N$Allele1.2==data_2017_N$Allele2.2 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1==data_2017_N$Allele1.2 & data_2017_N$Allele1=="NA" & data_2017_N$Allele2=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.1,
                ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.2==data_2017_N$Allele2.2 & data_2017_N$Allele2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele1.1=="NA" & data_2017_N$Allele2.1=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele2.2,"NA"))))))))))))))))))))

Allele1_N_2017 <- ifelse(data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2&
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
```

```
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1,
                ifelse(data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1,

               #ifelse ((data_2017_N$Allele1!=data_2017_N$Allele1.1 |
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 | data_2017_N$Allele1.2!=data_2017_N$Allele1) &
(data_2017_N$Allele2!=data_2017_N$Allele2.1 | data_2017_N$Allele2.1!=data_2017_N$Allele2.2 &
data_2017_N$Allele2!=data_2017_N$Allele2.2) & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", "NA",
                ifelse ((data_2017_N$Allele1!=data_2017_N$Allele1.1 &
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 & data_2017_N$Allele1.2!=data_2017_N$Allele1) &
(data_2017_N$Allele2!=data_2017_N$Allele2.1 | data_2017_N$Allele2.1!=data_2017_N$Allele2.2 &
data_2017_N$Allele2!=data_2017_N$Allele2.2) & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", "NA",

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2.2!=data_2017_N$Allele2 & data_2017_N$Allele2.1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2!=data_2017_N$Allele2.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1==data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.2 & data_2017_N$Allele2.1!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1.2,

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1.2!=data_2017_N$Allele1 & data_2017_N$Allele1.1==data_2017_N$Allele1 &
data_2017_N$Allele2.1==data_2017_N$Allele2 & data_2017_N$Allele2!="NA" & data_2017_N$Allele1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.2!="NA" &
data_2017_N$Allele1.2!="NA", data_2017_N$Allele1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele2.1==data_2017_N$Allele2.2 & data_2017_N$Allele2!="NA" & data_2017_N$Allele1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.2!="NA" &
data_2017_N$Allele1.2!="NA", data_2017_N$Allele1.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele1.1!=data_2017_N$Allele1.2 &
data_2017_N$Allele2!="NA" & data_2017_N$Allele1!="NA" & data_2017_N$Allele2.1!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1.2,

                ifelse(data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1.2,
                ifelse(data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1.1,
                ifelse(data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" & data_2017_N$Allele1.1!="NA" &
data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA",
data_2017_N$Allele1,

                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.2,
                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2!=data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.2 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.2,
                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1!=data_2017_N$Allele2.1 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.1,
                # ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele2==data_2017_N$Allele2.1 & data_2017_N$Allele1.1==data_2017_N$Allele1.2 &
data_2017_N$Allele1!=data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.1,
```

```
                # ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele2==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1,
                # ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele1.2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele2==data_2017_N$Allele2.1 &
data_2017_N$Allele1.1!=data_2017_N$Allele1.2 & data_2017_N$Allele1!="NA" & data_2017_N$Allele2!="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA" & data_2017_N$Allele1.2!="NA" &
data_2017_N$Allele2.2!="NA", data_2017_N$Allele1,

                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.1!=data_2017_N$Allele2.1 & data_2017_N$Allele2==data_2017_N$Allele2.1 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.2=="NA" & data_2017_N$Allele2.2=="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA", data_2017_N$Allele1,
                ifelse (data_2017_N$Allele1.1!=data_2017_N$Allele2.1 &
data_2017_N$Allele1.2!=data_2017_N$Allele2.2 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1==data_2017_N$Allele1.2 & data_2017_N$Allele1=="NA" & data_2017_N$Allele2=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.1,
                ifelse (data_2017_N$Allele1!=data_2017_N$Allele2 &
data_2017_N$Allele1.2!=data_2017_N$Allele2.2 & data_2017_N$Allele2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele1.1=="NA" & data_2017_N$Allele2.1=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.2,
                ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.1==data_2017_N$Allele2.1 & data_2017_N$Allele2==data_2017_N$Allele2.1 &
data_2017_N$Allele1==data_2017_N$Allele1.1 & data_2017_N$Allele1.2=="NA" & data_2017_N$Allele2.2=="NA" &
data_2017_N$Allele1.1!="NA" & data_2017_N$Allele2.1!="NA", data_2017_N$Allele1,
                ifelse (data_2017_N$Allele1.1==data_2017_N$Allele2.1 &
data_2017_N$Allele1.2==data_2017_N$Allele2.2 & data_2017_N$Allele2.1==data_2017_N$Allele2.2 &
data_2017_N$Allele1.1==data_2017_N$Allele1.2 & data_2017_N$Allele1=="NA" & data_2017_N$Allele2=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.1,
                ifelse (data_2017_N$Allele1==data_2017_N$Allele2 &
data_2017_N$Allele1.2==data_2017_N$Allele2.2 & data_2017_N$Allele2==data_2017_N$Allele2.2 &
data_2017_N$Allele1==data_2017_N$Allele1.2 & data_2017_N$Allele1.1=="NA" & data_2017_N$Allele2.1=="NA" &
data_2017_N$Allele1.2!="NA" & data_2017_N$Allele2.2!="NA", data_2017_N$Allele1.2,"NA"))))))))))))))))))


gt_N_17 <- cbind(data_2017_N,Allele1_N_2017,Allele2_N_2017)

gt_N_17[ gt_N_17 == "NA" ] <- NA


write.table(gt_N_17, file = "./Data/gt.wreps_17.txt", sep = "\t", row.names = F, col.names = T, quote=F)


new_DF_2017 <- gt_N_17[rowSums(is.na(gt_N_17)) > 0,] #shows where the missing values are

gt_tab_17.n <- gt_N_17[,c(1,2,9,10)]

# vis_dat(gt_tab_17.n)
#
# gg_miss_upset(gt_tab_17.n) #shows the distributions and combinations of missing data in the data frame
# gg_miss_var(gt_tab_17.n,
#             facet = Marker) #shows the missing values for the different markers -> Lsa1 has the most
missing values

gt_tab_17.n$Allele1_N_2017 <- as.numeric(as.character(gt_tab_17.n$Allele1_N_2017))
gt_tab_17.n$Allele2_N_2017 <- as.numeric(as.character(gt_tab_17.n$Allele2_N_2017))

plot(gt_tab_17.n$Allele2_N_2017~gt_tab_17.n$Allele1_N_2017)

gt_raw_17 <- gather(gt_tab_17.n, key = "k", value = "v", -SampleID, na.rm = F) %>%
  group_by(SampleID) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

gt.input_17 <- gt_raw_17[,c(1,2,9,3,10,4,11,5,12,6,13,7,14,8,15)]
colnames(gt.input_17) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2")       #rename coloumns

sex_17 <- read.table("./Data/results.sexdet_17.txt", header=T, na.strings = "NA")
compare(sex_17$SampleID, gt.input_17$SampleID)     #to make sure that they are the same and that the sex is
really for the same sample
gt.result_17 <- cbind(gt.input_17,sex_17$sex_17)

gt.clean_17 <- gt.result_17[-which(rowMeans(is.na(gt.result_17)) > 0.2), ]
#gt_clean.17$na_count <- apply(is.na(gt_clean.17), 1, sum)

colnames(gt.result_17) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2", "sex")       #rename
coloumns


write.table(gt.clean_17, file = "Data/genotypestable.clean_2017.txt", sep = "\t", row.names = F, col.names
= T, quote=F)
write.table(gt.result_17, file = "Data/genotypestable.all_2017.txt", sep = "\t", row.names = F, col.names =
T, quote=F)



#--------------------2018----------------------

data_2018_raw <- read.table("./Data/data_2018.txt", header=T, na.strings = "NA")
data_2018 <-
data.frame(data_2018_raw$SampleID,data_2018_raw$Marker,data_2018_raw$Allele1,data_2018_raw$Allele2)
#keep only coloumns that are needed
colnames(data_2018) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

#NA if data is ok
```

```
plot(data_2018$SampleID)

length(unique(data_2018$SampleID))
#[1] 345 -> there are 345 samples (faeces in the data of 2018)

nrow(data_2018)/length(unique(data_2018$SampleID))
#[1] 19.25507 -> not ok, unequal distribution of markers


#rearrange data so that all alleles are in one row
data_2018_NA <- gather(data_2018, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_2018_N <- data_2018_NA[,c(1,2,3,6,4,7,5,8)]
summary(data_2018_N)
plot(data_2018_N$Marker)

data_2018_N$Marker <- as.factor(data_2018_N$Marker)
summary(data_2018_N)

write.table(data_2018_N, file = "replicates data 2018.txt", sep = "\t", row.names = F, col.names = T,
quote=F)


# vis_dat(data_2018_N)
#
# gg_miss_upset(data_2018_N) #shows the distributions and combinations of missing data in the data frame ->
the 4th replication is missing in 280 markers, the there are 182 times only two replicates per sample and
marker
# gg_miss_var(data_2018_N,
#             facet = Marker) #shows the missing values for the different markers -> Sol30 has the most
missing values

# ggplot(data_2018_N,
#        aes(x = Allele1.2,
#            y = Allele1.1)) +
#   geom_miss_point() +
#   facet_wrap(~Marker)


#change NA to "NA" -> not missing value but characters

# data_2018_N$Allele1 <- as.numeric(as.character(data_2018_N$Allele1))
# data_2018_N$Allele2 <- as.numeric(as.character(data_2018_N$Allele2))
# data_2018_N$Allele1.1 <- as.numeric(as.character(data_2018_N$Allele1.1))
# data_2018_N$Allele2.1 <- as.numeric(as.character(data_2018_N$Allele2.1))
# data_2018_N$Allele1.2 <- as.numeric(as.character(data_2018_N$Allele1.2))
# data_2018_N$Allele2.2 <- as.numeric(as.character(data_2018_N$Allele2.2))

data_2018_N[is.na(data_2018_N)] <- "NA"

Allele2_N_2018 <- ifelse (data_2018_N$Allele1==data_2018_N$Allele2 &
data_2018_N$Allele1.1==data_2018_N$Allele2.1 & data_2018_N$Allele1.2==data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele1==data_2018_N$Allele1.1 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA", data_2018_N$Allele2,
                          ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele1==data_2018_N$Allele1.1 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2.1,

                                  #ifelse ((data_2018_N$Allele1!=data_2018_N$Allele1.1 |
data_2018_N$Allele1.1!=data_2018_N$Allele1.2 | data_2018_N$Allele1.2!=data_2018_N$Allele1) &
(data_2018_N$Allele2!=data_2018_N$Allele2.1 | data_2018_N$Allele2.1!=data_2018_N$Allele2.2 &
data_2018_N$Allele2!=data_2018_N$Allele2.2) & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", "NA",
                                  ifelse ((data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1.1!=data_2018_N$Allele1.2 & data_2018_N$Allele1.2!=data_2018_N$Allele1) &
(data_2018_N$Allele2!=data_2018_N$Allele2.1 | data_2018_N$Allele2.1!=data_2018_N$Allele2.2 &
data_2018_N$Allele2!=data_2018_N$Allele2.2) & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", "NA",

                                        ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2.2!=data_2018_N$Allele2 & data_2018_N$Allele2.1==data_2018_N$Allele2 &
data_2018_N$Allele1.1==data_2018_N$Allele1 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2,
                                                ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2!=data_2018_N$Allele2.1 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.1,
                                                        ifelse (data_2018_N$Allele1!=data_2018_N$Allele2
& data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele2.1!=data_2018_N$Allele2.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2.2,

                                                                ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1.2!=data_2018_N$Allele1 &
```

```
data_2018_N$Allele1.1==data_2018_N$Allele1 & data_2018_N$Allele2.1==data_2018_N$Allele2 &
data_2018_N$Allele2!="NA" & data_2018_N$Allele1!="NA" & data_2018_N$Allele2.1!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.2!="NA" & data_2018_N$Allele1.2!="NA",
data_2018_N$Allele2,
                                                                    ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele2!="NA" & data_2018_N$Allele1!="NA" & data_2018_N$Allele2.1!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.2!="NA" & data_2018_N$Allele1.2!="NA",
data_2018_N$Allele2.1,
                                                                        ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele1.1!=data_2018_N$Allele1.2 & data_2018_N$Allele2!="NA" & data_2018_N$Allele1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.2,

                                                                            ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.2,
                                                                                ifelse
(data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.1,

ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2,


ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.2,

ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.1,

ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2,


# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2.2,

# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele2==data_2018_N$Allele2.2 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2.2,

# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2.1,

# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele2==data_2018_N$Allele2.1 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele1!=data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2.1,

# ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele1.1!=data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2,

# ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele1.1!=data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele2,


ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele1.2=="NA" & data_2018_N$Allele2.2=="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA", data_2018_N$Allele2,

ifelse (data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
```

```
data_2018_N$Allele1=="NA" & data_2018_N$Allele2=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.1,

ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele1.1=="NA" & data_2018_N$Allele2.1=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.2,

ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele1.2=="NA" & data_2018_N$Allele2.2=="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA", data_2018_N$Allele2,

ifelse (data_2018_N$Allele1.1==data_2018_N$Allele2.1 & data_2018_N$Allele1.2==data_2018_N$Allele2.2 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
data_2018_N$Allele1=="NA" & data_2018_N$Allele2=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.1,

ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.2==data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele1.1=="NA" & data_2018_N$Allele2.1=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele2.2,"NA"))))))))))))))))))))

Allele1_N_2018 <- ifelse(data_2018_N$Allele1==data_2018_N$Allele2 &
data_2018_N$Allele1.1==data_2018_N$Allele2.1 & data_2018_N$Allele1.2==data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele1==data_2018_N$Allele1.1 & data_2018_N$Allele1.1==data_2018_N$Allele1.2&
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1,
                    ifelse(data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele1==data_2018_N$Allele1.1 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1,

                                #ifelse ((data_2018_N$Allele1!=data_2018_N$Allele1.1 |
data_2018_N$Allele1.1!=data_2018_N$Allele1.2 | data_2018_N$Allele1.2!=data_2018_N$Allele1) &
(data_2018_N$Allele2!=data_2018_N$Allele2.1 | data_2018_N$Allele2.1!=data_2018_N$Allele2.2 &
data_2018_N$Allele2!=data_2018_N$Allele2.2) & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", "NA",
                                ifelse ((data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1.1!=data_2018_N$Allele1.2 & data_2018_N$Allele1.2!=data_2018_N$Allele1) &
(data_2018_N$Allele2!=data_2018_N$Allele2.1 | data_2018_N$Allele2.1!=data_2018_N$Allele2.2 &
data_2018_N$Allele2!=data_2018_N$Allele2.2) & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", "NA",

                                        ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2.2!=data_2018_N$Allele2 & data_2018_N$Allele2.1==data_2018_N$Allele2 &
data_2018_N$Allele1.1==data_2018_N$Allele1 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1,
                                                ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2!=data_2018_N$Allele2.1 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.1,
                                                        ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 &
data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele2.1!=data_2018_N$Allele2.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1.2,

                                                                ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1.2!=data_2018_N$Allele1 &
data_2018_N$Allele1.1==data_2018_N$Allele1 & data_2018_N$Allele2.1==data_2018_N$Allele2 &
data_2018_N$Allele2!="NA" & data_2018_N$Allele1!="NA" & data_2018_N$Allele2.1!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.2!="NA" & data_2018_N$Allele1.2!="NA",
data_2018_N$Allele1,
                                                                        ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele2.1==data_2018_N$Allele2.2 &
data_2018_N$Allele2!="NA" & data_2018_N$Allele1!="NA" & data_2018_N$Allele2.1!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.2!="NA" & data_2018_N$Allele1.2!="NA",
data_2018_N$Allele1.1,
                                                                                ifelse
(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele1.1!=data_2018_N$Allele1.2 & data_2018_N$Allele2!="NA" & data_2018_N$Allele1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.2,


ifelse(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.2,

ifelse(data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.1,
```

```
ifelse(data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" &
data_2018_N$Allele1.1!="NA" & data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1,


# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1.2,

# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2!=data_2018_N$Allele2.2 & data_2018_N$Allele2==data_2018_N$Allele2.2 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele1!=data_2018_N$Allele1.1 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1.2,

# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1.1,

# ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele2==data_2018_N$Allele2.1 &
data_2018_N$Allele1.1==data_2018_N$Allele1.2 & data_2018_N$Allele1!=data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1.1,

# ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele1.1!=data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1,

# ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele1.2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele1.1!=data_2018_N$Allele1.2 &
data_2018_N$Allele1!="NA" & data_2018_N$Allele2!="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA" & data_2018_N$Allele1.2!="NA" & data_2018_N$Allele2.2!="NA",
data_2018_N$Allele1,


ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.1!=data_2018_N$Allele2.1 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele1.2=="NA" & data_2018_N$Allele2.2=="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA", data_2018_N$Allele1,

ifelse (data_2018_N$Allele1.1!=data_2018_N$Allele2.1 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
data_2018_N$Allele1=="NA" & data_2018_N$Allele2=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.1,

ifelse (data_2018_N$Allele1!=data_2018_N$Allele2 & data_2018_N$Allele1.2!=data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele1.1=="NA" & data_2018_N$Allele2.1=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.2,

ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.1==data_2018_N$Allele2.1 &
data_2018_N$Allele2==data_2018_N$Allele2.1 & data_2018_N$Allele1==data_2018_N$Allele1.1 &
data_2018_N$Allele1.2=="NA" & data_2018_N$Allele2.2=="NA" & data_2018_N$Allele1.1!="NA" &
data_2018_N$Allele2.1!="NA", data_2018_N$Allele1,

ifelse (data_2018_N$Allele1.1==data_2018_N$Allele2.1 & data_2018_N$Allele1.2==data_2018_N$Allele2.2 &
data_2018_N$Allele2.1==data_2018_N$Allele2.2 & data_2018_N$Allele1.1==data_2018_N$Allele1.2 &
data_2018_N$Allele1=="NA" & data_2018_N$Allele2=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.1,

ifelse (data_2018_N$Allele1==data_2018_N$Allele2 & data_2018_N$Allele1.2==data_2018_N$Allele2.2 &
data_2018_N$Allele2==data_2018_N$Allele2.2 & data_2018_N$Allele1==data_2018_N$Allele1.2 &
data_2018_N$Allele1.1=="NA" & data_2018_N$Allele2.1=="NA" & data_2018_N$Allele1.2!="NA" &
data_2018_N$Allele2.2!="NA", data_2018_N$Allele1.2,"NA")))))))))))))))))


gt_N_18 <- cbind(data_2018_N,Allele1_N_2018,Allele2_N_2018)

gt_N_18[ gt_N_18 == "NA" ] <- NA


write.table(gt_N_18, file = "./Data/gt.wreps_18.txt", sep = "\t", row.names = F, col.names = T, quote=F)


new_DF_2018 <- gt_N_18[rowSums(is.na(gt_N_18)) > 0,] #shows where the missing values are

gt_tab_18.n <- gt_N_18[,c(1,2,9,10)]

# vis_dat(gt_tab_18.n)
#
# gg_miss_upset(gt_tab_18.n) #shows the distributions and combinations of missing data in the data frame
# gg_miss_var(gt_tab_18.n,
#             facet = Marker) #shows the missing values for the different markers -> Lsa1 has the most
missing values

gt_tab_18.n$Allele1_N_2018 <- as.numeric(as.character(gt_tab_18.n$Allele1_N_2018))
gt_tab_18.n$Allele2_N_2018 <- as.numeric(as.character(gt_tab_18.n$Allele2_N_2018))
```

```
plot(gt_tab_18.n$Allele2_N_2018~gt_tab_18.n$Allele1_N_2018)

gt_raw_18 <- gather(gt_tab_18.n, key = "k", value = "v", -SampleID, na.rm = F) %>%
  group_by(SampleID) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

gt.input_18 <- gt_raw_18[,c(1,2,9,3,10,4,11,5,12,6,13,7,14,8,15)]
colnames(gt.input_18) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2")       #rename coloumns

sex_18 <- read.table("./Data/results.sexdet_18.txt", header=T, na.strings = "NA")
compare(sex_18$SampleID, gt.input_18$SampleID)     #to make sure that they are the same and that the sex is
really for the same sample
gt.result_18 <- cbind(gt.input_18,sex_18$sex_18)

gt.clean_18 <- gt.result_18[-which(rowMeans(is.na(gt.result_18)) > 0.2), ]
#gt_clean.18$na_count <- apply(is.na(gt_clean.18), 1, sum)

colnames(gt.result_18) <- c("SampleID", "Lsa1.1", "Lsa1.2", "Lsa3.1", "Lsa3.2", "Sat5.1", "Sat5.2",
"Sat8.1","Sat8.2", "Sol30.1", "Sol30.2", "Sol33.1", "Sol33.2", "Sol8.1", "Sol8.2", "sex")       #rename
coloumns


write.table(gt.clean_18, file = "Data/genotypestable.clean_2018.txt", sep = "\t", row.names = F, col.names
= T, quote=F)
write.table(gt.result_18, file = "Data/genotypestable.all_2018.txt", sep = "\t", row.names = F, col.names =
T, quote=F)
```

Appendix 1.2: Table with conditions for acceptance of consensus genotypes

Table 14: Example for the conditions for acceptance of multilocus genotypes. Alleles assumed as correct are given in green, alleles assumed to be "wrong" are given in red.

Original replicate Alleles						**Resulting Alleles**	
Allele 1	Allele 2	Allele 1.1	Allele 2.1	Allele 1.2	Allele 2.2	Allele 1	Allele 2
171	171	171	171	171	171	171	171
171	175	171	175	171	175	171	175
171	175	171	171	175	175	NA	NA
206	208	208	210	210	210	NA	NA
171	175	171	175	171	210	171	175
171	210	171	175	171	175	171	175
171	175	171	210	171	175	171	175
171	175	171	175	150	175	171	175
150	175	171	175	171	175	171	175
171	175	150	175	171	175	171	175
171	175	171	171	171	175	171	175
171	171	171	175	171	175	171	175
171	175	171	175	171	171	171	175
171	175	171	175	NA	NA	171	175
NA	NA	171	175	171	175	171	175
171	175	NA	NA	171	175	171	175

Appendix 1.3: Sex determination in R based on replicates

```
#DATA INPUT = table with two alleles for SRY marker -> if both alleles are absent, could be female but does not have to be
#both alleles present = male

library(dplyr)
library(tidyr)

#--------------2014-------------------
sry.2014 <- read.table("./Data/data_2014_SRY.txt", header=T, na.strings = "NA")
sry.2014 <- data.frame(sry.2014$SampleID,sry.2014$Marker,sry.2014$Allele1,sry.2014$Allele2)    #keep only coloumns that are needed
colnames(sry.2014) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

data_sex.14 <- gather(sry.2014, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID, Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_sex.14 <- data_sex.14[,c(1,2,3,6,4,7,5,8)]
str(data_sex.14)

sex.14 <- ifelse((is.na(data_sex.14$Allele1)) & (is.na(data_sex.14$Allele1.1)) & (is.na(data_sex.14$Allele1.2)),"female", "male")
sex_only.14 <- data.frame(data_sex.14$SampleID,sex.14)
colnames(sex_only.14) <- c("SampleID", "sex")       #rename coloumns

sex_all.14 <- cbind(data_sex.14,sex.14)
write.table(sex_all.14, file = "./Data/results.sexdet_14.txt", sep = "\t", row.names = F, col.names = T, quote=F)

#-----------------2015----------------
sry.2015 <- read.table("./Data/data_2015_SRY.txt", header=T, na.strings = "NA")
sry.2015 <- data.frame(sry.2015$SampleID,sry.2015$Marker,sry.2015$Allele1,sry.2015$Allele2)    #keep only coloumns that are needed
colnames(sry.2015) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

data_sex.15 <- gather(sry.2015, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_sex.15 <- data_sex.15[,c(1,2,3,6,4,7,5,8)]

sex_15 <- ifelse((is.na(data_sex.15$Allele1)) & (is.na(data_sex.15$Allele1.1)) & (is.na(data_sex.15$Allele1.2)),"female", "male")
sex_only.15 <- data.frame(data_sex.15$SampleID,sex_15)
colnames(sex_only.15) <- c("SampleID", "sex")       #rename coloumns

sex_all.15 <- cbind(data_sex.15,sex_15)
sex_all.15$sex <- ifelse(sex_all.15$sex_15=="female",0,1)
summary(sex_all.15$sex_15)    #108 female samples classified, 168 male samples

write.table(sex_all.15, file = "./Data/results.sexdet_15.txt", sep = "\t", row.names = F, col.names = T, quote=F)


#-----------------2016----------------
sry.2016 <- read.table("./Data/data_2016_SRY.txt", header=T, na.strings = "NA")
sry.2016 <- data.frame(sry.2016$SampleID,sry.2016$Marker,sry.2016$Allele1,sry.2016$Allele2)    #keep only coloumns that are needed
colnames(sry.2016) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

data_sex.16 <- gather(sry.2016, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
```

```
  ungroup()

data_sex.16 <- data_sex.16[,c(1,2,3,6,4,7,5,8)]

sex_16 <- ifelse((is.na(data_sex.16$Allele1)) & (is.na(data_sex.16$Allele1.1)) &
(is.na(data_sex.16$Allele1.2)),"female", "male")
sex.only_16 <- data.frame(data_sex.16$SampleID,sex_16)
colnames(sex.only_16) <- c("SampleID", "sex")       #rename coloumns

sex.all_16 <- cbind(data_sex.16,sex_16)
sex.all_16$sex <- ifelse(sex.all_16$sex_16=="female",0,1)
summary(sex.all_16$sex_16)    #178 male samples, 132 female

write.table(sex.all_16, file = "./Data/results.sexdet_16.txt", sep = "\t", row.names = F, col.names = T,
quote=F)



#-----------------2017----------------
sry.2017 <- read.table("./Data/data_2017_SRY.txt", header=T, na.strings = "NA")
sry.2017 <- data.frame(sry.2017$SampleID,sry.2017$Marker,sry.2017$Allele1,sry.2017$Allele2)    #keep only
coloumns that are needed
colnames(sry.2017) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

data_sex.17 <- gather(sry.2017, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_sex.17 <- data_sex.17[,c(1,2,3,6,4,7,5,8)]

sex_17 <- ifelse((is.na(data_sex.17$Allele1)) & (is.na(data_sex.17$Allele1.1)) &
(is.na(data_sex.17$Allele1.2)),"female", "male")
sex.only_17 <- data.frame(data_sex.17$SampleID,sex_17)
colnames(sex.only_17) <- c("SampleID", "sex")       #rename coloumns

sex.all_17 <- cbind(data_sex.17,sex_17)
sex.all_17$sex <- ifelse(sex.all_17$sex_17=="female",0,1)
summary(sex.all_17$sex_17)    #126 male samples, 202 female

write.table(sex.all_17, file = "./Data/results.sexdet_17.txt", sep = "\t", row.names = F, col.names = T,
quote=F)



#-----------------2018----------------
sry.2018 <- read.table("./Data/data_2018_SRY.txt", header=T, na.strings = "NA")
sry.2018 <- data.frame(sry.2018$SampleID,sry.2018$Marker,sry.2018$Allele1,sry.2018$Allele2)    #keep only
coloumns that are needed
colnames(sry.2018) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

data_sex.18 <- gather(sry.2018, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID,Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data_sex.18 <- data_sex.18[,c(1,2,3,6,4,7,5,8)]

sex_18 <- ifelse((is.na(data_sex.18$Allele1)) & (is.na(data_sex.18$Allele1.1)) &
(is.na(data_sex.18$Allele1.2)),"female", "male")
sex.only_18 <- data.frame(data_sex.18$SampleID,sex_18)
colnames(sex.only_18) <- c("SampleID", "sex")       #rename coloumns

sex.all_18 <- cbind(data_sex.18,sex_18)
sex.all_18$sex <- ifelse(sex.all_18$sex_18=="female",0,1)

summary(sex.all_18$sex_18)    #164 male samples, 181 female

write.table(sex.all_18, file = "./Data/results.sexdet_18.txt", sep = "\t", row.names = F, col.names = T,
quote=F)
```

Appendix 1.4: R Script for finding unique genotypes (allelematch)

```
library(dplyr)
library(tidyr)
library(allelematch)



#read in data for sry loci
sry.2014 <- read.table("./Data/data_2014_SRY.txt", header=T, na.strings = "NA")
sry.2015 <- read.table("./Data/data_2015_SRY.txt", header=T, na.strings = "NA")
sry.2016 <- read.table("./Data/data_2016_SRY.txt", header=T, na.strings = "NA")
sry.2017 <- read.table("./Data/data_2017_SRY.txt", header=T, na.strings = "NA")
sry.2018 <- read.table("./Data/data_2018_SRY.txt", header=T, na.strings = "NA")
sry.2014$SampleID <- as.factor(sry.2014$SampleID)
sry.2015$SampleID <- as.factor(sry.2015$SampleID)
sry.2016$SampleID <- as.factor(sry.2016$SampleID)
sry.2017$SampleID <- as.factor(sry.2017$SampleID)
sry.2018$SampleID <- as.factor(sry.2018$SampleID)


#read in supplementary information
samples <- read.table("./Data/samples1418.txt", header = T)
samples$SampleID <- as.factor(samples$SampleID)

#read genotype data for all years and combine to one file
d14 <- read.table("./Data/genotypestable.all_2014.txt", header = T, na.strings = "NA")
d15 <- read.table("./Data/genotypestable.all_2015.txt", header = T, na.strings = "NA")
d16 <- read.table("./Data/genotypestable.all_2016.txt", header = T, na.strings = "NA")
d17 <- read.table("./Data/genotypestable.all_2017.txt", header = T, na.strings = "NA")
d18 <- read.table("./Data/genotypestable.all_2018.txt", header = T, na.strings = "NA")

d14$SampleID <- as.factor(d14$SampleID)
d15$SampleID <- as.factor(d15$SampleID)
d16$SampleID <- as.factor(d16$SampleID)
d17$SampleID <- as.factor(d17$SampleID)
d18$SampleID <- as.factor(d18$SampleID)

allgenotypes <- rbind(d14,d15,d16,d17,d18)
allgt <- allgenotypes[order(match(allgenotypes$SampleID, samples$SampleID)),]

#combine sex data with supplementary information and extract only neccessary information (make one genotype
from 3 replicates)
sry.all_raw <- rbind(sry.2014, sry.2015, sry.2016, sry.2017, sry.2018)

sry.all_1 <- data.frame(sry.all_raw$SampleID,sry.all_raw$Marker,sry.all_raw$Allele1,sry.all_raw$Allele2)
#keep only coloumns that are needed
colnames(sry.all_1) <- c("SampleID", "Marker", "Allele1", "Allele2")       #rename coloumns

data.sry <- gather(sry.all_1, key = "k", value = "v", -SampleID, -Marker) %>%
  group_by(SampleID, Marker) %>%
  mutate(k = make.unique(k)) %>%
  spread(key = k, value = v) %>%
  ungroup()

data.sry <- data.sry[c(1,2,3,6,4,7,5,8)]
sry.fin <- data.sry[order(match(data.sry$SampleID, samples$SampleID)),]
data.sry$SampleID <- as.factor(data.sry$SampleID)

#comp_1 <- c(samples$SampleID[!samples$SampleID %in% data.sry$SampleID])
#write.table(comp_1, file = "./Data/missing_samples_18.txt", sep = "\t", row.names = F, col.names = T,
quote=F)

#samples_new <- samples[-c(comp_1), ]
s#amples_new$SampleID <- as.integer(as.character(samples_new$SampleID))
sry.fin$SampleID <- as.integer(as.character(sry.fin$SampleID))

comp <- sry.fin$SampleID==samples_new$SampleID

summary(comp)   #they are the same

sry.full <- cbind(sry.fin,samples)
sry.all <- sry.full[c(1:8,10,11)]

sex.A1 <- ifelse((is.na(sry.all$Allele1)) & (is.na(sry.all$Allele1.1)) & (is.na(sry.all$Allele1.2)),"0",
"299")
sex.A2 <- ifelse((is.na(sry.all$Allele1)) & (is.na(sry.all$Allele1.1)) & (is.na(sry.all$Allele1.2)),"0",
"299")

sex_all <- cbind(sry.all,sex.A1,sex.A2)

sry <- sex_all[,c(11,12)]
colnames(sry) <- c("Sry.1", "Sry.2")


#comine everything to one data file
data.full <- cbind(allgt,samples,sry)
data.full <- data.full[c(1:15,23,24,16,18:21)]

write.table(data.full, file = "./Data/gt_table_with all info.txt", sep = "\t", row.names = F, col.names =
T, quote=F)
write.table(data.full, file = "./Data/gt_table_suppinfo_sry.txt", sep = "\t", row.names = F, col.names = T,
quote=F)
```

```
data.full_clean <- data.full[-which(rowMeans(is.na(data.full)) > 0.15), ]
write.table(data.full_clean, file = "./Data/results/gt_table_suppinfo_sry_clean.txt", sep = "\t", row.names
= F, col.names = T, quote=F)


data.full_clean$na_count <- apply(is.na(data.full_clean), 1, sum)

#create table for allelematch
data.match <- data.full[c(1:18)]


#allelematch
data.match <- data.match[-which(rowMeans(is.na(data.match)) > 0.2), ]

unique.all <- amDataset(data.match, indexColumn="SampleID", missingCode=NA, metaDataColumn = "sex")
amUniqueProfile(unique.all, doPlot=TRUE)

result.full_0 <- amUnique(unique.all, alleleMismatch=0)
result.full_2 <- amUnique(unique.all, alleleMismatch = 2)

summary(result.full_0, html="./Data/results/results.sry.0.html" )
summary(result.full_2, html="./Data/results/results.sry.2.html" )
summary(result.full_0, csv="./Data/results/results.sry_0.csv")
summary(result.full_2, csv="./Data/results/results.sry_2.csv")



#genotyping all samples, without sry loci

data.7loci  <- data.match[-c(16,17)]

unique.7 <- amDataset(data.7loci, indexColumn="SampleID", missingCode=NA, metaDataColumn = "sex")
amUniqueProfile(unique.7, doPlot=TRUE)

result.7l_0 <- amUnique(unique.7, alleleMismatch=0)
result.7l_2 <- amUnique(unique.7, alleleMismatch = 2)

summary(result.7l_0, html="./Data/results/results.7l.0.html" )
summary(result.7l_2, html="./Data/results/results.7l.2.html" )
summary(result.7l_0, csv="./Data/results/results.7l_0.csv")
summary(result.7l_2, csv="./Data/results/results.7l_2.csv")
```

Appendix 1.5: Examples of peaks of additional loci

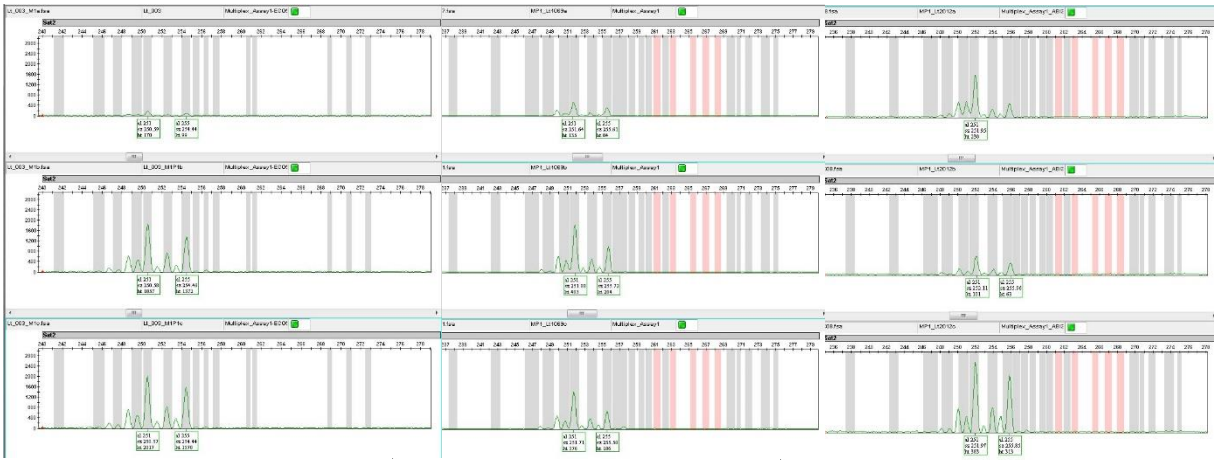

Figure 17: Examples of the phenotype found for Sat2 for individual ID1, shown as example by 3 samples.

Figure 18: Examples of the phenotype found for Sat2 for individual ID20, shown as example by 3 samples.

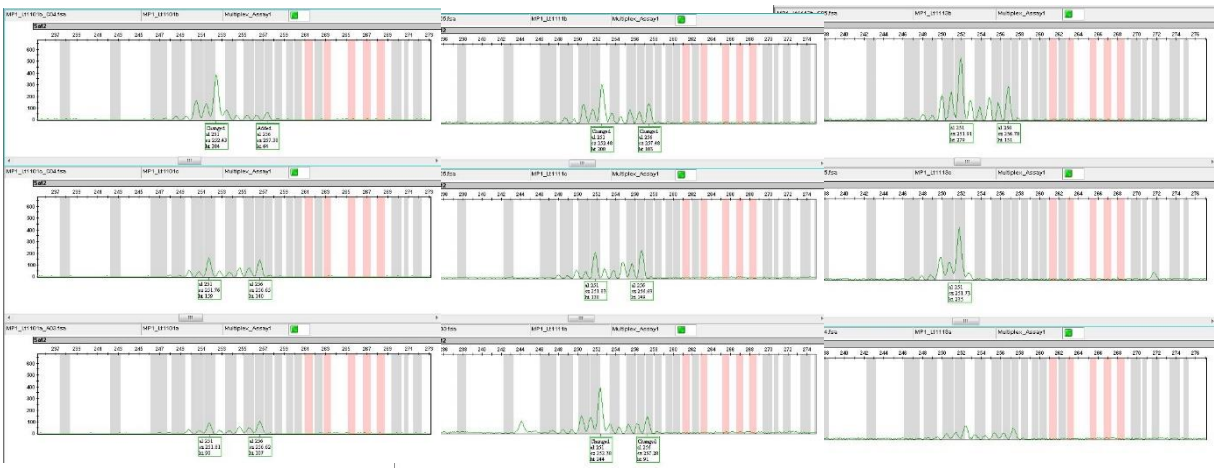

Figure 19: Examples of the phenotype found for Sat2 for individual ID60, shown as example by 3 samples.

Appendix 2: Additional information to pedigree analysis with COLONY

Table 15: Overview of the number of clusters obtained using different parameter values and data inputs.

NC	NC08	N_{CO}	N_{CM}	N_{CF}	E	P_P
3	0	71	15	11	high	0.9
5	2	71	15	11	high	0.5
6	3	71	15	11	low	0.9
5	2	71	15	11	low	0.5
5	0	59	10	3	high	0.9
5	0	59	10	3	high	0.5
5	1	59	10	3	low	0.9
6	1	59	10	3	low	0.5
4	0	40	16	5	high	0.9
4	0	40	16	5	high	0.5
4	1	40	16	5	low	0.9
4	1	40	16	5	low	0.5
4	0	24	10	2	high	0.9
6	2	24	10	2	high	0.5
5	0	24	10	2	low	0.9
4	1	24	10	2	low	0.5
3	0	10	9	6	high	0.9
3	0	10	9	6	high	0.5
3	0	10	9	6	low	0.9
3	0	10	9	6	low	0.5

Appendix 3: Correlation between the number of samples and the number of individuals identified in each sampling session

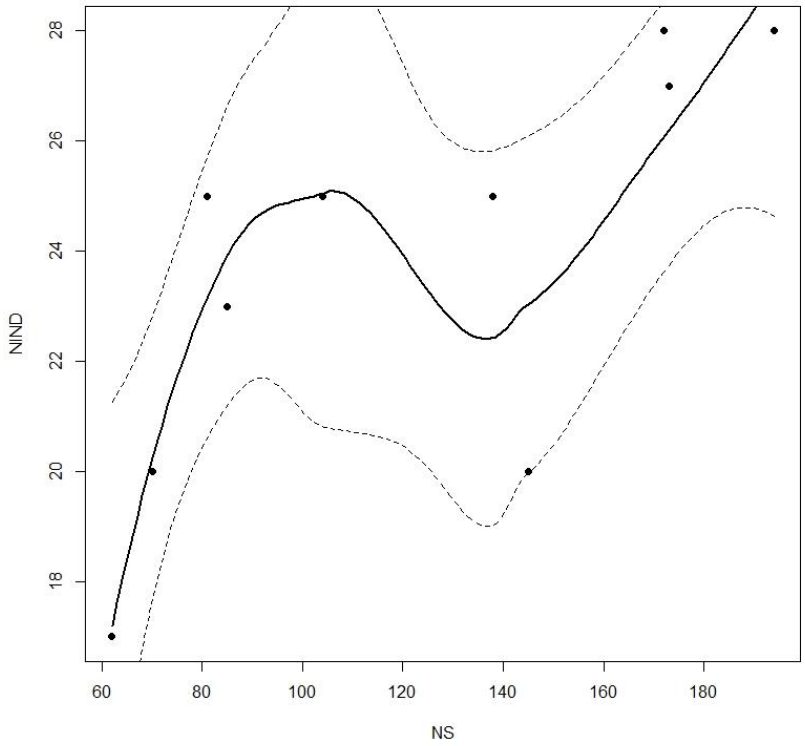

Figure 20: Relationship between the number of samples (NS) and the number of individuals detected (NIND), including a 95% Confidence Interval.