Distinction between evolutionary scenarios using supervised machine learning

Pierrot Van der Aa

October 2023 - June 2024

Université Libre de Bruxelles

Mémoire présenté en vue de l'obtention du diplôme de

Master en Bioinformatique et Modélisation

Promotors: Prof. Dr. Patrick Mardulyn and Prof. Dr. Tom Lenaerts

Abstract

Phylogeography, the study of the correlation between individual genotypes and spatial distribution can be done through the comparison of the observed data with simulation data under defined evolutionary scenarios. One approach to tackle this kind of question is called the Approximate Bayesian Computation (ABC). However, this requires simulating a large number of data and selecting the appropriate descriptors of those data (summary statistics). In addition to the computational burden of producing simulations for datasets larger and larger, finding the right summary statistics is a trade-off between too many leading to a curse of dimensionality and too few meaning a loss of information. In recent years, new approaches have been proposed to avoid those problems. They use a random forest approach, which is a supervised machine learning algorithm. In this thesis, we dive deeper in the use of machine learning tools to answer evolutionary questions. Our research shows that those tools can be used to distinguish evolutionary scenarios which are nonlinearly separable. We then applied our pipeline to a real-life example of leaf beetle populations in European mountains (*Gonioctena quinquepunctata*) and obtained mixed results requiring further investigations.

Table of content

Contents

Abstract

Table of content

1 Introduction **1**

1.1 Phylogeography in a computational context 1

1.2 Population genetics in the machine learning era 4

1.3 Research context 5

2 Goals **8**

3 Material and methods **9**

3.1 Production of simulated data 9

3.2 Selection of the summary statistics 14

3.3 Linearity of the data 15

3.4 Selection of the ML algorithms 16

3.4.1 Dummy classifier 19

3.4.2 Logistic regression 19

3.4.3 Decision tree 19

3.4.4 Random forest 20

3.4.5 Histogram-based gradient boosting classification tree 21

3.5 Life sample 21

4 Results **22**

4.1 General results 22

4.2 Accuracy 23

4.3 Confusion matrices 24

4.4 ROC curves 24

4.5 Other metrics 27

4.6 Permutation importance 27

4.7 Life sample 28

5 Discussion **30**

5.1 Computing time 30

5.2 Comparison with other techniques 31

5.3 Most relevant features 31

5.4 Life sample 32

6 Conclusion **34**

Acknowledgments **35**

References **36**

Appendices **44**

1 Introduction

1.1 Phylogeography in a computational context

Phylogeography is the field of population genetics which studies genetic variation to infer evolutionary history of species and their geographical distribution ranges (Avise 2000). Studying evolutionary history requires to compute simulations and compare them with the observed data. Indeed, evolutionary history is too complex to be inferred only by fossils data. It requires to model the stochasticity of natural process. This is where simulations become useful because they allow to explore different combinations of parameters encompassing the natural variability of the evolutionary history (Hoban, Bertorelle, and Gaggiotti 2012). These simulations can be done in two different ways: either forward or backward in time. The forward simulations follow the life cycle of each individual and therefore allow for more complex models but are more computationally heavy. The backward simulations on the other hand look at the lineage and are based on the coalescent theory (Kingman 1982) which is a model describing the evolution of alleles within populations. According to the coalescent theory, each allele can be traced back in time to a common ancestor (i.e. the most recent common ancestor (MRCA), see Figure 1). This model is then extended to all alleles of the individuals from a given population. Two components are calculated by the coalescent: the coalescent time (time needed for two alleles to trace back to the MRCA) and the parameter θ ($4N_e\mu$ where N_e is the effective population size and μ is the mutation rate) (Sigwart 2009). The model makes four assumptions:

1. that there is no recombination of the alleles (breaking of chromosomes which leads to combinations of alleles different from those of the parental generation),
2. that the allele is not under natural selection pressure,
3. that there is no gene flow (also referred as migration),
4. that there is no population structure (i.e. that there is random mating).

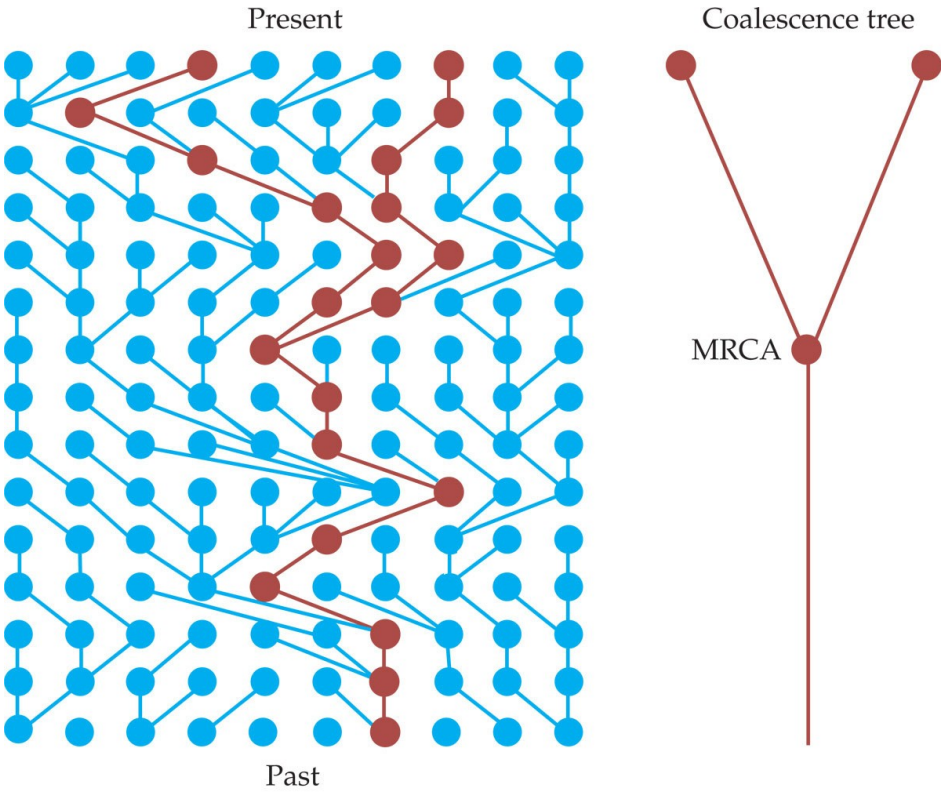

Figure 1: Illustration of a gene genealogy. Two individuals are sampled in the current population and by tracing back in time their history, we can find the most recent common ancestor (MRCA) thanks to the coalescent theory (Nielsen and Slatkin 2013).

These strong assumptions ensure that each allele has the same probability to be passed from one generation to the next. Comparing the observed data to the simulated ones is done via the use of the likelihood function (i.e. the probability of the observations given the data). Some software allows to compute this likelihood function for small population numbers (e.g.: 4 populations maximum in the case of δaδi (Gutenkunst 2009)) but these software cannot handle complex scenarios of evolutionary history nor a big amount of genetic data (Excoffier et al. 2013). The likelihood function is, in most cases intractable or computationally too heavy. For this reason, a couple of techniques have arisen to approximate the likelihood function or avoid it entirely. We will present the different techniques presented in Nielsen and Beaumont (2009) but will focus mainly on the Approximate Bayesian Computation (ABC) method:

Methods without mutation: one work around is to consider that there are no mutations. This is possible if the mutation rate is low and is exceeded by the migration rate. The calculations are simplified because it only relies on gene frequencies allowing to compute the likelihood function (Nielsen et al. 1998).

Product of approximating conditionals (PAC): coalescent methods estimate the parameter θ while this method mainly applies to looking for recombination rates by considering multiple loci at the same time instead of looking at pairs of loci (N. Li and Stephens 2003).

Composite likelihood: in this approach, more simple likelihood estimates are done on subsets of the dataset and then aggregated into a composite function (Hudson 2001). This method assumes that all the subsets are independent (Larribe and Fearnhead 2011). An advantage of this approach is that it is not much impacted by the size of the genetic sample considered (Excoffier et al. 2021). Therefore, this method follow efficiently the current trend of increasing -omic data. The composite likelihood is often used in combination with a site frequency spectrum (SFS) which describes the distribution of allele frequency in the populations (Nielsen 2000). To name a few programs using the composite likelihood: `fastsimcoal2` (Excoffier et al. 2013) and `momi2` (Kamm et al. 2020).

Approximate Bayesian Computation (ABC) (Rubin 1984): this last approach, also called likelihood-free, starts by computing a vast amount of simulations under different parameter

sets. It then computes summary statistics which are numerical descriptions of genetic data in terms of variability and differentiation (see 'Material and methods' for examples). The Euclidian distance between the summary statistics vector of the observed data and the ones from the simulated data is then computed. The samples close enough to the observed data according to the rejection sampling criterion are retained to describe the parameter distributions (Tavaré et al. 1997; Pritchard et al. 1999; Beaumont, Zhang, and Balding 2002; Marjoram et al. 2003). The drawbacks of the ABC method are the computational load in the case of large -omic data and the need to select a set of sufficiently descriptive summary statistics (Pudlo et al. 2016). The latter could be solved by adding as many summary statistics as possible but the rejection sampling phase suffers from the curse of dimensionality (Blum et al. 2013). To solve the problem of the choice of summary statistics, multiple ideas have been proposed. The first one is to simply discard summary statistics which bring little information. The second one consists in adding weights to the different summary statistics to avoid rejecting completely the ones carrying few information. The last option is to use dimension reduction techniques such as PLS (Partial Least Squares), LDA (Linear Discriminant Analysis) or PCA (Principal Component Analysis) (Beaumont 2010). `DIYABC` (Cornuet et al. 2014) and `ABCtoolbox` (Wegmann et al. 2010) are examples of software using the ABC approach.

The following thesis focuses on recent improvements of the ABC techniques to overcome the obstacles mentioned above.

1.2 Population genetics in the machine learning era

Recently, the progress made in the field of machine learning has met the field of population genetics. The first interaction between the two fields can be found in Boitard, Schlötterer, and Futschik (2009) who used hidden Markov models (HMM) to detect selective sweeps (fixation of a mutation under selection in a population). Still on selective sweeps, the following year, another paper was published using support vector machines (SVM) this time (Pavlidis, Jensen, and Stephan 2010). From that period onward, the number of papers linking population genetics and machine learning kept growing (e.g. Sheehan (2016); Schrider and Kern

(2016); Chapuis et al. (2020); Smith and Carstens (2020)). They rely on the supervised approach of machine learning in which a training set is provided to an estimator. In the case of classification, the set of variables (called features) is associated with a target variable containing the classes that the model must 'recognise'. The model performance is then evaluated on a test set of data not previously seen by the algorithm. One algorithm has been particularly popular and is called Random Forest (RF) (Breiman 2001). Fernández-Delgado et al. (2014) compared 179 classifiers on 112 datasets and showed that three random forest algorithms landed in the top five of the accuracy leaderboard. This algorithm is based on the aggregation of multiple decision trees (see 'Material and methods'). The random forest approach has been applied to the ABC method because it allows to circumvent its drawbacks. Indeed, the ABC requires to produce a big number of simulations from which a certain number is discarded by the rejection sampling procedure (see description of ABC above) (i.e. will be rejected the simulations which have a too big Euclidian distance to the observation) but the machine learning approach can keep all the simulations and use them to train the algorithm. This allows for the use of a smaller dataset of simulations as a training set for machine learning (Schrider and Kern 2018). The advantage of using random forest compared to other algorithms is that it deals well with a big feature space by using subsamples of the features to construct the decision trees. Therefore, this solves the issue of the curse of dimensionality caused by the number of summary statistics in ABC (Pudlo et al. 2016; Schrider and Kern 2018). Thus, the combination of simulation data via ABC and classification by a random forest gave rise to new ABC-RF algorithms, which are faster, more robust and less computationally heavy (Pudlo et al. 2016; Fraimout et al. 2017; Raynal et al. 2019; Collin et al. 2021).

1.3 Research context

In this thesis, we will propose alternatives and improvements to the current ABC-RF software available (i.e. `abcrf`(Pudlo et al. 2016; Raynal et al. 2019)) and `DIYABC Random Forest` (Collin et al. 2021)). Indeed, the articles describing the software do not mention if their program performs hyperparameter tuning. This important step of machine learning consists

in changing the parameters that are describing the machine learning model and that are not learned by the model during its learning phase. As an example, in the case of a random forest, the number of trees that are part of the ensemble as well as their depth are important hyperparameters (Scornet 2017). The number of trees is the only one mentioned in the ABC-RF software article from Pudlo et al. (2016); it is said that the number of trees is set to 500. However, tuning the hyperparameters should be done for each dataset independently and not fixed once and for all (Andonie 2019). This is why we propose to add a two-step hyperparameter tuning in our model (see 'Material and methods'). The second difference between our approach and the ones mentioned above is that we implement four different algorithms (i.e. logistic regression, decision tree, random forest and histogram-based gradient boosting classification tree), allowing us to compare them. Finally, we use `fastsimcoal2` to simulate our genetic data. This program is faster (Excoffier et al. 2021) and allows for complex migration matrices not permitted by `DIYABC`.

Our study will focus on the classification of evolutionary scenarios of *Gonioctena quinquepunctata* (Fabricius 1787) (Coleoptera, Chrysomelidae). This leaf beetle has been described in 1787 by Fabricius (1787). It is a montane species, meaning that it can live in altitude ranging from 600 to 1300m and thus in between the temperate and the alpine altitudes (Schmitt, Muster, and Schönswetter 2010). It lives on host plants of the species *Prunus padus* and the genus *Sorbus* (Quinzin and Mardulyn 2014). Recent studies have tried to distinguish between different evolutionary scenarios based on RADseq data (Kastally et al. submitted). They explored different variations of two main hypotheses. The first one (lowland refuge hypothesis) states that the species colonised the lower altitude during the last glacial maximum therefore expanding its range. The second one (peripheral refuge hypothesis) presents what is called a glacial contraction meaning that the population reduced in size and that the insects were limited to small regions at the lowest range of their montane distribution. The study concludes that the information brought by the RADseq data is more in favour of the second hypothesis whereas species distribution models (SDM) simulations seem to rather go for the first hypothesis. Therefore, further sampling has been done and a full genome has been assembled (Lukicheva, Flot, and Mardulyn 2021). We now have 69 full genomes of different localisation throughout the distribution range of the species (the Alps,

Pyrenees, Massif Central and Vosges) as displayed in Figure 2.

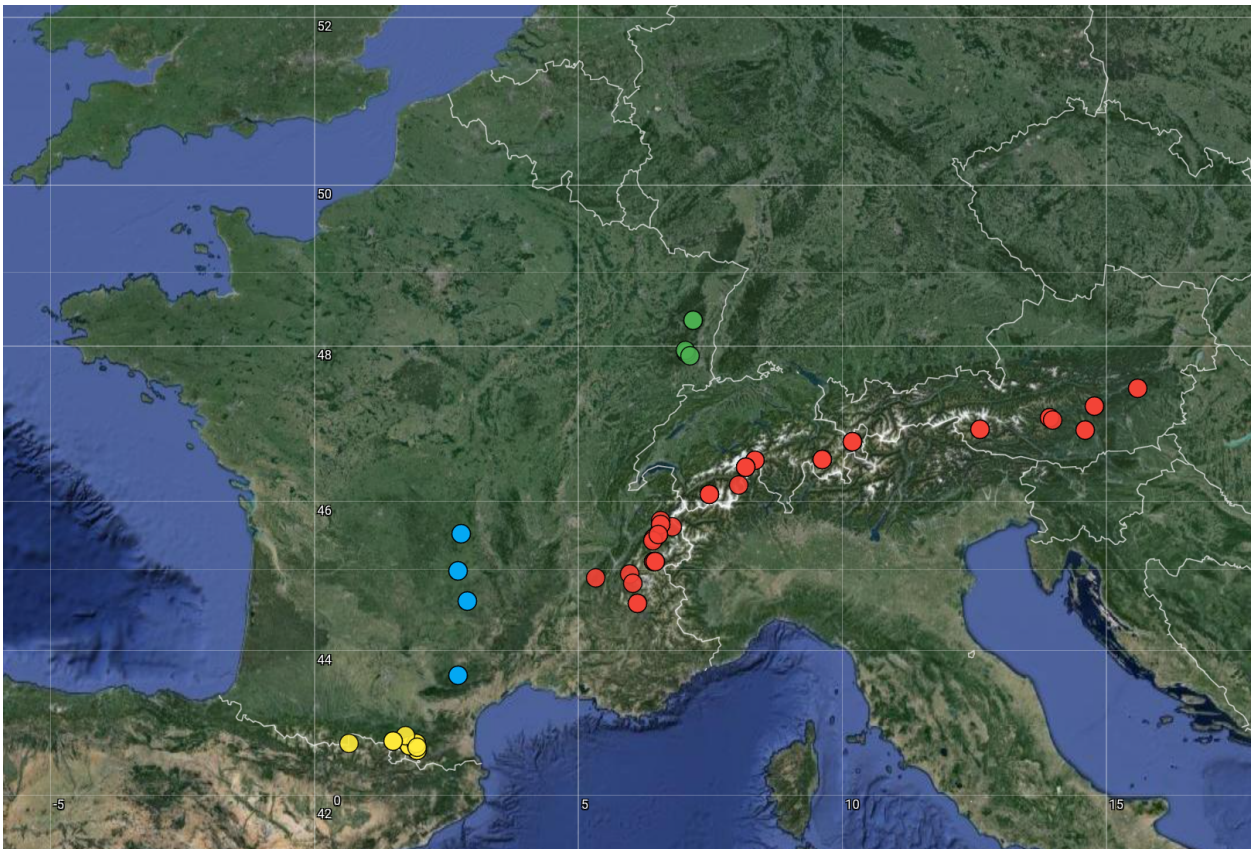

Figure 2: Map of the different sampling locations for the 69 full genomes: 45 from the Alps (red), 5 from the Vosges (green), 8 from the Massif Central (blue) and 11 from the Pyrenees (yellow).

2 Goals

We identified two objectives for this project.

The first and main one is to assess whether or not a machine learning algorithm distinguishes different evolutionary scenarios. Indeed, we want to assess if the simulations produced with `fastsimcoal2` under the specified scenarios can be classified by a multi-class classifier into four categories corresponding to the four different evolutionary scenarios that we will define in the 'Material and methods'. Given that we try different algorithms and another simulation pipeline, we need to evaluate if our framework is capable of handling the problem at hand. If so, it would mean that we could translate the ABC-RF approach to other algorithms and use a different tool for data simulations.

The second objective, if our algorithms can distinguish the four different scenarios, is to use them to predict in which evolutionary scenario real-life data would fall into. Indeed, we do now have genomic data from the current living populations of *Gonioctena quinquepunctata* and we would like to use them to trace back their evolutionary history.

3 Material and methods

3.1 Production of simulated data

The first step of the work was the production *in silico* of DNA sequences using the coalescent-based program `fastsimcoal 2.7` (Excoffier et al. 2013). This program, among other features, allows to produce synthetic DNA data on the basis of an input file specifying the evolutionary history of the populations considered. It is more suited for the production of genomic data than its predecessor, `ms`, more suited for smaller genetic data (Hudson 2002). In this thesis, four different scenarios were explored. They were selected based on previous research conducted on the same topic (Kastally et al. submitted) and because we expected them to be easily distinguishable by the machine learning algorithms due to their heterogeneity. They can be described as follow:

- Hypothesis00: fragmentation: in this scenario, the size of the four populations considered (one per mountain chain) did not vary in time. The panmictic initial population splits into four different populations whose sizes are proportional to the mountain chain area considered. The size of the populations stays the same during the entire glaciation until the present time.
- Hypothesis01: lowland refuge: according to this scenario, during the last glacial era, the different populations of *G. quinquepunctata* would have gathered in a lower altitude region as a panmictic population. At the end of the glacial era, with the increase of the temperature, different individuals would have colonised the mountain chains leading to a so-called founder effect (selection at random of a subset of alleles in the pool of the original population). This means that there is a loss of diversity during the founder effect.
- Hypothesis02: peripheral refuge: this scenario proposes that the four populations stayed in the mountains at the lower edge of their distribution range during the last glacial era but that the size of the populations reduced in size and remained small during the whole glacial era and increased at the end of the cold period to reach their current size. The smaller population size during the glacial era leads to a loss of

diversity due to stronger genetic drift.

- Hypothesis03: peripheral refuge with recolonisation from the Alps: in this last hypothesis, similar to Hypothesis02, the whole panmictic population was reduced in size during the last glacial era and the colonisation of the Vosges, the Pyrenees and the Massif Central was made from the population of the Alps. The genetic diversity decreases due to smaller population size during the glacial era but also during the recolonisation of the other mountain chains because of a founder effect.

All the scenarios presented on Figure 4 display a decrease in population size at the most ancient time. This allows for coalescence in `fastsimcoal`. Indeed, if the populations are too big, it is hard to find the MRCA. Reducing the population size after the relevant historical events allows therefore to, in fine, find the MRCA.

Defining those scenarios was done using both a template provided with `fastsimcoal` and data from parallel analyses of other researcher in the lab working on model parameters estimations on the basis of SFS. Each analysis requires two files: a `.tpl` template file and a `.est` file for estimation of parameters (they can be found in Appendices 1, 2, 3 and 4). The template files are structured as follows:

- The first block of inputs defines the effective population size of each of our four populations. We set it as a parameter, meaning that we gave a range of values to the program in which it would sample a set of values for each run done (i.e. between 1000 and 10000 for the Alps). The range was given for the Alps population and we consider that the size of the populations of the other mountain chains were proportional to that of the Alps population (i.e. 0.25 times the population of the Alps for the Massif Central, 0.1657 for the Pyrenees and 0.0335 for the Vosges). These numbers correspond to the montane area (between 600 and 1300m of altitude) available in those different mountain chains.
- The second block allows us to define the number of sampled individuals in each population. In our case, we selected 45 individuals from the Alps, 5 from the Vosges, 11 from the Pyrenees and 8 from the Massif Central. Indeed, these numbers fit what the lab had been sampling in the wild.

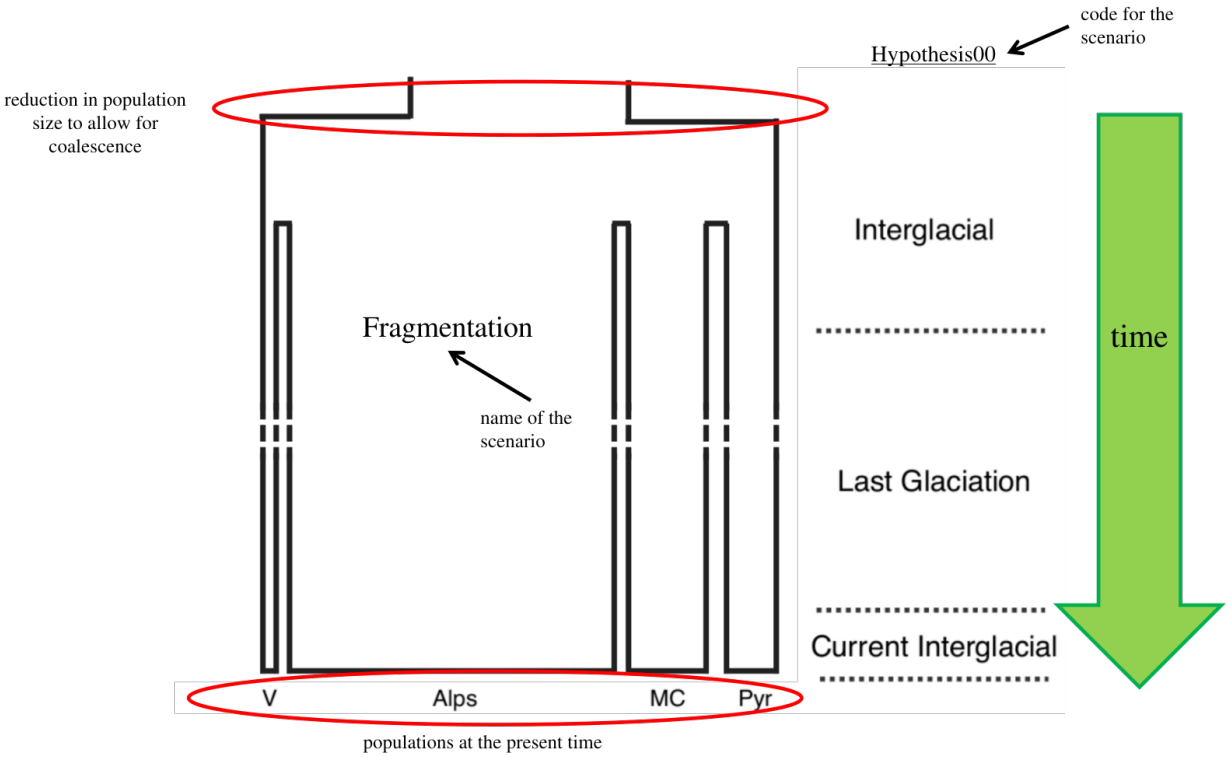

Figure 3: This illustration dissects the structure of a given evolutionary scenario to help understand Figure 4. Each scenario has a code (top right) used throughout the thesis to be referenced but also a short name to describe the scenario (center). The top of the figure represent the most ancient time where we reduce the population size to allow the coalescence to happen. Time goes from most recent, present time (bottom), to more ancient time (top) because `fastsimcoal` works backward in time. There is a geological scale on the right indicating when the different evolutionary events occurred. The bifurcations on this drawing indicate that the panmictic ancestral population split into different populations whose names are abbreviated at the bottom of the drawing (V = Vosges, MC = Massif Central and Pyr = Pyrenees). The width of the branches is proportional to the effective population size.

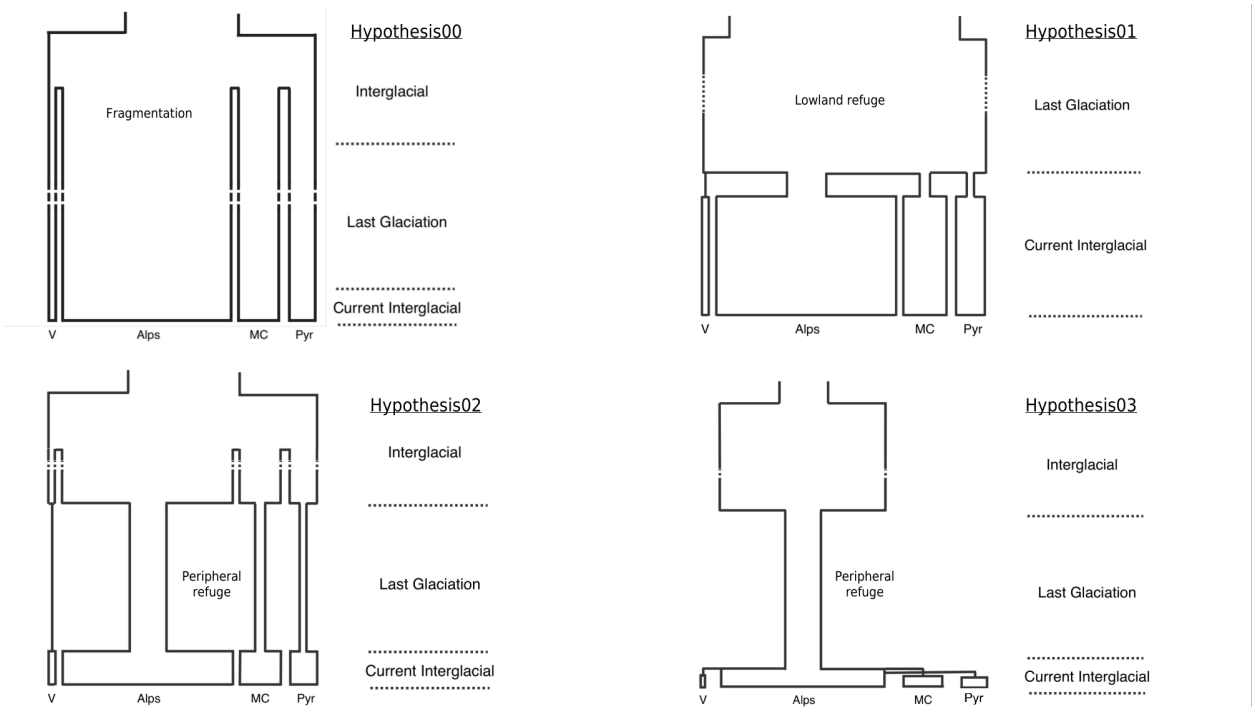

Figure 4: Schematic representation of the different evolutionary scenarios explored with `fastsimcoal` (adapted from Kastally et al. (submitted)). Hypothesis00 denotes a simple fragmentation of the panmictic population in the four mountain chains. Hypothesis01 displays the panmictic population in low altitude during the last glacial era with a founder effect while recolonising the mountain chains. Hypothesis02 shows that, during the last glacial era, the populations stayed in the mountains in reduced population size at the fringe of their distribution (peripheral refuge). Hypothesis03 also displays a peripheral refuge hypothesis for the Alps with a recolonisation of the other mountain chains from the population of the Alps (V = Vosges, MC = Massif Central and Pyr = Pyrenees).

- The third and fourth block were set to zero to simplify the analyses. They are meant to invite the user to put a growth rate of the current population and a migration matrix.
- The fifth block is the one dedicated to encoding historical events as described above. For Hypothesis00, the parameters were the time of fragmentation (between 10000 and 200000 years) and the time for the resize of the population (computed as a number between 10 and 200000 plus the time of fragmentation). For Hypothesis01, the parameters were the time for the bottleneck (corresponding to the beginning of the Holocene between 10000 and 12000 years ago), the time for the resize of the population (corresponding to the beginning of the Last Glacial Period between 100000 and 120000 years) and the strength of the bottleneck (calculated as one divided by the number of individuals, itself calculated as a proportion of the initial population with a factor ranging from 1e-5 to 1e-1). For Hypothesis02, the parameters were the time of the historical events (see above), the resize factors for the population decrease at the beginning of the glacial period (between 1e-4 and 1e-1) and for the population increase at the end of the glacial period (between 1 and 1e4). Finally, for Hypothesis03, the parameters were the time of the historical events, the size increase and decrease (see parameters of Hypothesis02) and the bottleneck strength (calculated as before with bottleneck strength ranging from 1e-5 to 1e-2). For all hypotheses, the strength of the resize of the initial population to allow coalescence was set between 1 and 100.
- The last blocks are dedicated to the description of the DNA dataset to produce. We considered 20000 independent SNPs (Single Nucleotide Polymorphisms), knowing that this number may vary depending on the convergence of the simulation, with a mutation rate of 2.1e-7 and a recombination rate of 0.33, which implies no transition bias. Given that the programmer of `fastsimcoal2` advises not to produce SNPs as such but rather short fragments of DNA, we computed 20000 fragments, all of which were long of 10 base pairs. This number of 20000 independent SNPs aligns with the genomic data that we have from the real-life sample. The mutation rate, μ, was tuned together with the effective population size, N_e, in order to end up with enough mutations in the files to have a genetic signal but not too many mutations which would not be biologically relevant.

Given that a certain number of descriptors of the input files were set as variable parameters (i.e. the time at which the historical events were occurring, the strength of population resize, the effective population size and the resize factor after bottleneck event), a set of ranges were defined in the `.est` file. This variability means that each run of the program uses a different set of parameters sampled at random. Therefore, it is important to repeat the analysis multiple times. It was decided that 10000 independent runs per scenarios would be done which fits the proposed number of iterations in Pudlo et al. (2016). Thus, the final dataset would include 40000 datapoints and would be composed of balanced classes which will ease the machine learning analyses.

We ran the first `fastsimcoal2` trials and the summary statistics analyses (see further) on the CECI computer cluster.

`Fastsimcoal2` outputs files in Arlequin format `.arp`. However, this format is mainly used by the software Arlequin and arlsumstat (Excoffier and Lischer 2010) which were designed for rather small datasets at the genetic era. Therefore, it appeared important to use programs which were both more recent and designed for larger datasets to reduce the computational time and load on the cluster. Given that these alternatives were not able to handle the Arlequin format, it was necessary to first perform a step of file conversion to the widely used Variant Call Format (`.vcf`) (Danecek et al. 2011). This conversion was performed using the software PGDSpider 2.1.1.5 (Lischer and Excoffier 2012). Some software also required that the VCF file was compressed and indexed using bcftools 1.19 (H. Li 2011).

3.2 Selection of the summary statistics

The SNP data produced by `fastsimcoal` were summarised using so-called summary statistics which describe DNA sequence variation within and among populations. A comprehensive list of those summary statistics can be found in Table 1. They were produced using the free and open source programming language R version 4.1.2 (R Core Team 2021) and the following packages: PopGenome 2.7.5 (Pfeifer et al. 2014), vcfR 1.15.0 (Knaus and Grünwald 2017), mmod 1.3.3 (Winter 2012), hierfstat 0.5.11 (Goudet and Jombart 2015), adegenet 2.1.10 (Jombart 2008; Jombart and Ahmed 2011) and dartR 2.9.7 (Gruber et al. 2018;

Table 1: Summary statistics used in this thesis and the reference in which they have been described. When there were multiple instances of a same summary statistics but for different populations or combination of populations, the populations numbers were replaced in the table by X and Y in order to have a more compact table.

Abbreviation	R.package	Summary.statistics	Source
Hs_PopX	vcfR	average subpopulation Hardy–Weinberg heterozygosity for popX	Hedrick and Philip (2005)
Ht	vcfR	total population heterozygosity	Hedrick and Philip (2005)
Gst	vcfR	genetic differenciation Gst = (Ht-Hs)/Ht	Hedrick and Philip (2005)
Htmax	vcfR	maximum heterozygosity possible in the total population	Hedrick and Philip (2005)
Gstmax	vcfR	maximum genetic differenciation possible in the total population	Hedrick and Philip (2005)
Gprimest	vcfR	Gst' = Gst/Gstmax	Hedrick and Philip (2005)
a	vcfR	see equation 13 of Jost 2008	Jost (2008)
b	vcfR	see equation 13 of Jost 2008	Jost (2008)
Dest_Chao	vcfR	unibiased estimator of D based on a and b, D=1-(a/b)	Jost (2008)
Fst	hierfstat	the correlation of genes of different individuals in the same population (coancestry)	Weir and Cockerham (1984)
Fis	hierfstat	the correlation of genes within individuals within populations	Weir and Cockerham (1984)
Hs	mmod	expected heterozygosities under Hardy-Weinberg equilibrium or gene diversities within subpopulations	Nei and Chesser (1983)
Ht.1	mmod	expected heterozygosities under Hardy-Weinberg equilibrium or gene diversities in the total	Nei and Chesser (1983)
Gst_est	mmod	genetic differenciation	Nei and Chesser (1983)
Gprime_st	mmod	global estimtes for Gst' based on overall heterozygosity	Hedrick and Philip (2005)
D_het	mmod	D based on overall heterozigosity	Jost (2008)
D_mean	mmod	Harmonic mean of values of D per locus	Jost (2008)
Phi_st	mmod	standardized measure of genetic differentiation under the AMOVA framework	Meirmans (2005)
Ho	hierfstat	observed heterozygosity	Nei (1987)
Hs.1	hierfstat	within population gene diversity	Nei (1987)
Ht.2	hierfstat	overall gene diversity	Nei (1987)
Dst	hierfstat	gene diversity among samples	Nei (1987)
Htp	hierfstat	Ht'=Hs+Dst'	Nei (1987)
Dstp	hierfstat	Dst'=np/(np-1)Dst	Nei (1987)
Fst.1	hierfstat	Dst/Ht	Nei (1987)
Fstp	hierfstat	Dst'/Ht'	Nei (1987)
Fis.1	hierfstat	1-Ho/Hs	Nei (1987)
Dest	hierfstat	Dest=np/(np-1) (Ht'-Hs)/(1-Hs) a measure of population differentiation	Jost (2008)
Mean_kinship	hierfstat	pairwise kinship and individual inbreeding coefficients	Weir and Goudet (2017)
dist_Dch_PopX_Y	hierfstat	genetic distance between subpopulations based on the Dch method	Takezaki and Nei (1996)
dist_Da_PopX_Y	hierfstat	genetic distance between subpopulations based on the Da method	Takezaki and Nei (1996)
dist_Ds_PopX_Y	hierfstat	genetic distance between subpopulations based on the Ds method	Takezaki and Nei (1996)
beta_PopX_Y	hierfstat	pairwise kinship and individual inbreeding coefficients for pairs of populations	Weir and Goudet (2017)
neifst_PopX_Y	hierfstat	pairwise Fsts according to Nei (1987)	Nei (1987)
WCfst_PopX_Y	hierfstat	pairwise Fsts according to Weir and Cockerham (1984)	Weir and Cockerham (1984)
pi	hierfstat	nucleotide diversity	Nei and Li (1979)
TajimaD	hierfstat	statistic to test the neutral mutation hypothesis (Tajima's D)	Tajima (1989)
theta	hierfstat	Estimation of the product of the effective population size and the neutral mutation rate theta=4 Ne μ	Watterson (1975)
pi_popX	hierfstat	nucleotide diversity of pop1	Nei and Li (1979)
TajimaD_popX	hierfstat	statistic to test the neutral mutation hypothesis (Tajima's D) per population	Tajima (1989)
dxy_PopX_Y	PopGenome	absolute nucleotide divergence between two populations	Nei (1987)

Mijangos et al. 2022).

The idea was to compute as many summary statistics as possible and let the machine learning discriminate the most relevant features. We ended up having 84 different features.

3.3 Linearity of the data

We explored three different techniques to see if our dataset was linearly separable:

- Principal component analysis (PCA) (Pearson 1901): this technique allows the visualisation of high-dimensional data into a 2D space. It is a linear technique that reduces the dimensionality of the data while capturing the maximum of the variance of the

data. If the classification task is linearly separable, the datapoints may cluster distinctly and we would be able to draw a line between the different clusters with our real-life data being part of one of them.

- t-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton 2008): similarly to the PCA, t-SNE is a dimension reduction technique allowing to see relationships in the data in a lower dimensional space. However, this is a nonlinear technique preserving the distance between datapoints. If our classification task is separable, the datapoints will form distinct clusters.
- Logistic regression: By using a linear model in our machine learning pipeline, we can asses if the data is linearly separable. Indeed, if the task is not linearly separable, the algorithm will not converge.

3.4 Selection of the ML algorithms

A good rule of thumb when it comes to selecting machine learning algorithms is to start with a simple linear model and, if the problem is not linearly separable, to explore further other algorithms (Pedregosa et al. 2011). Therefore, the first classifier we explored was the logistic regression. We then further explored decision trees and other tree based methods (random forest and histogram-based gradient boosting classification tree) because they are well suitable for tabular data (Amor et al. 2023). Given that we were building a multi-class classifier, we focused on algorithms which natively support this approach (Aly 2005; Pedregosa et al. 2011). All the analyses were performed using functions from the `scikit-learn` package version 1.3.0 (Pedregosa et al. 2011) in the programming language python version 3.11.5 (Van Rossum and Drake 2009). All classifiers were evaluated with an inner 5-fold cross-validation for the evaluation of the hyperparameters and outer 10-fold cross-validation for the evaluation of the model performance on the test set (see Figure 5 for a graphical representation). For the metric to optimise when doing the cross-validation, we decided to select the accuracy. Given that we had a dataset with balanced classes, the accuracy denotes well the proportion of well classified samples. The dataset was split into 50% training set and 50% test set to preserve a good balance between generalisation and

overfitting. The search for optimal hyperparameters was done in two steps. The first grid search was performed on a broad grid of values with ranges inspired from trials made on a small toy dataset (Anderson 1936; Fisher 1936). The second grid search, was done around the optimal value returned by the first fitting. All the data were first fed through a standard scaler, which subtracts the mean and divide by the standard deviation. This step is not compulsory for tree-based algorithm but can improve the computational time (Genuer et al. 2017). Along with the accuracy, our main metric to evaluate the models, other metrics were used to evaluate the different models:

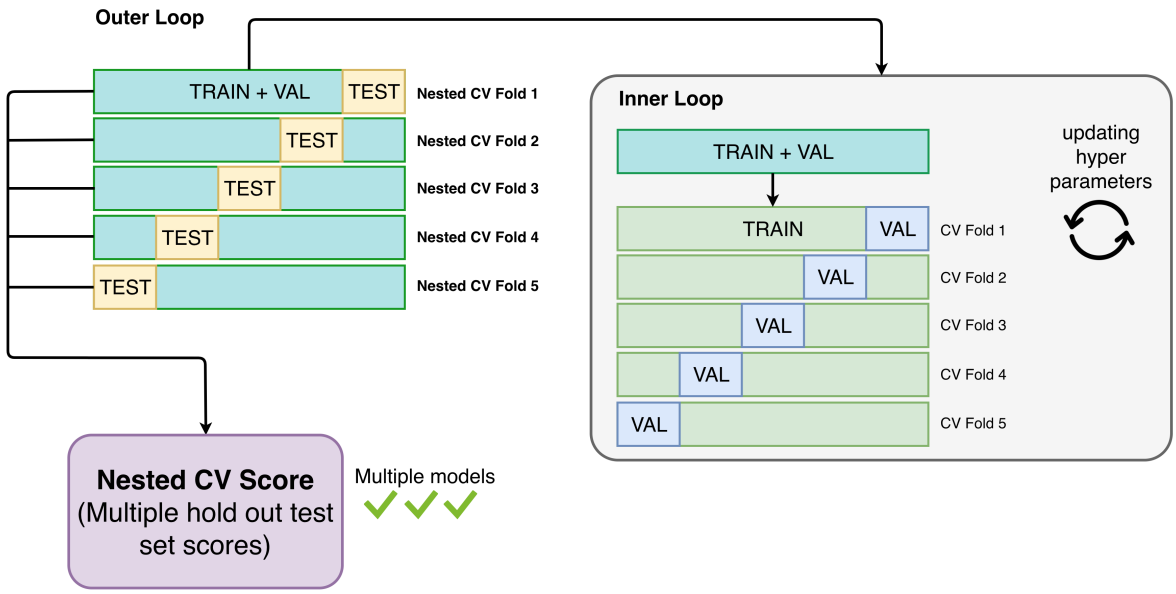

Figure 5: Illustration of the cross-validation used in this thesis (HackingMaterials 2019). The main dataset is split in ten train-test sets (only five on the drawing) in the outer loop. Each of the train sets from the outer loop are then split five times into train and validation sets in the inner loop. The inner loop aims to fine-tune the hyperparameters while the outer loop is for the model evaluation.

- confusion matrix: this matrix returns the true label of the target variable on the rows and the predicted labels from the model on the columns. On the diagonal of this matrix, we can find the number of times an object has been correctly classified, those are called the true positive (TP). We want to have as many samples from the test set on this diagonal as possible.

- precision: this is the ratio between TP and the sum of TP and false positive (FP). Precision measures the proportion of true positive predictions among all positive predictions made by the model. In other words, a high precision means that when the classifier predicts a positive class, it is likely to be right.
- recall: this is the ratio between TP and the sum of TP and false negative (FN). Recall measures the proportion of samples classified as positive among all the positive samples in the target variable. In other words, a high recall means that the FN number is low.
- f1-score: there's a trade-off between precision and recall but the f1-score allows to combine those two metrics as it is the weighted harmonic mean of the two.
- ROC curves and AUC: receiver operating characteristic curves are graphs which display the false positive rate on the x axis and the true positive rate on the y axis. The diagonal of this plot denotes a random classifier with efficacy of 50%, so not better than random. The performance of a binary classifier is plotted on the graph at varying thresholds. Ideally, the false positive rate should be close to zero and the true positive rate close to one. The performance of a classifier can be computed by the area under the ROC curve (AUC). The ROC curves are designed for binary classification but our work is a multi-class classification problem. We can overcome this issue by the approach of one-vs-rest (Allwein, Schapire, and Singer 2000). In this approach, the performance of the classifier is plotted for a given class against the others. The calculations are repeated with a one-vs-rest approach for each of the class.

These classification metrics require that the user defines a certain threshold. Given that we had no a priori knowledge on a class being better or more important, this threshold to distinguish one class from another was the majority.

A permutation importance test was performed to assess the importance of each feature. This technique allows to retrieve feature importance based on the data. Feature importance could be retrieve from the random forest directly but the permutation importance allows to expand the technique to any algorithms (Altmann et al. 2010). This is why we decided to use this approach to better compare the results from the different classifiers in terms of feature importance. The permutation importance shuffles the variables of a feature and assesses the

impact of the operation on the model error. If the difference in model error compared to the non-shuffled feature is big, it means that the feature is important.

To support the reproducibility of the machine learning method of this study, the machine learning summary table (Appendix 5) is included in the supporting information as per DOME recommendations (Walsh et al. 2021).

3.4.1 Dummy classifier

A dummy classifier sets a baseline for the exploration of more complex classifiers. Indeed, this classifier ignores the input space and returns values based on a given strategy. In our case, we selected the strategy 'uniform' which returns values at random while preserving the class balance. Given that we have four classes, this classifier has an accuracy of 25%. A classifier with a higher accuracy is thus considered more relevant than this baseline random classifier.

3.4.2 Logistic regression

Logistic regression is a type of linear model (Cox 1958). It makes the assumption that the dataset is linearly separable and thus looks for the hyperplane separating the classes in the feature space dimension. Linear models are regarded as simpler models and this is the reason why we start by evaluating this model before moving on to more complex models. For this classifier, the hyperparameter that we fine-tuned via a two-layer grid-search was the regularisation strength, C. We used `scikit-learn` `LogisticRegression` function with 10000 maximum iterations.

3.4.3 Decision tree

A decision tree (Quinlan 1986) partitions the space of input variables into subsets and keeps doing that recursively until all the samples in a subset belong to a same target variable. It is composed of nodes splitting into two according to a condition being true or false. Every end of the tree is called a leaf (see Figure 6). A fully split tree can overfit the data.

Therefore, the extension step of the tree is followed by a step of pruning which will reduce the size of the tree and keep the best one according to the cost complexity pruning. The hyperparameter fine-tuned for our decision tree was the depth of the tree. The function `DecisionTreeClassifier` from `scikit-learn` was used for this algorithm.

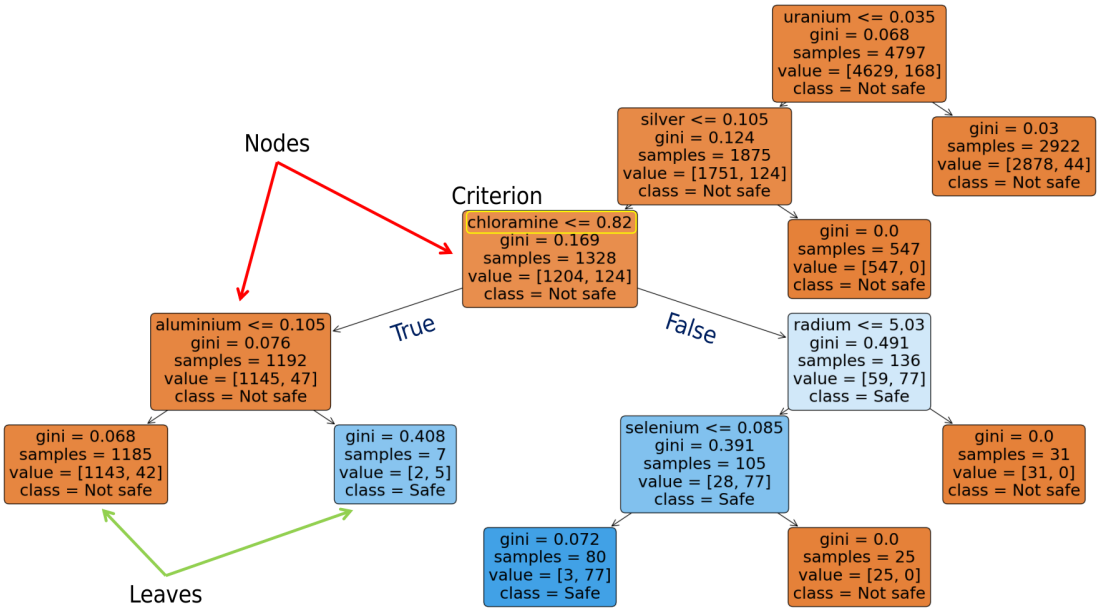

Figure 6: Example of a pruned binary decision tree with 6 nodes and 7 leaves constructed on a water quality dataset. In this case, there are only two classes: safe (drinkable) and not safe (not drinkable). The first line in each node is the split criterion.

3.4.4 Random forest

A random forest (Breiman 2001) is a bagging (bootstrap + aggregating) approach in which a set of non-pruned trees are computed on subsets of the feature space. These trees alone have low bias, but might overfit the data. They are then averaged to reduce the variance of single trees. This is the aggregation step. The hyperparameters, in this case, are the number of trees and the depth of those trees. We used the `RandomForestClassifier` function from `scikit-learn`.

3.4.5 Histogram-based gradient boosting classification tree

This boosting method (Schapire 1990) consists in iteratively building a tree based on a set of weak learners (small trees). Given that the tree is built iteratively, the computation time can be high. This is why we opted for the histogram-based (HGBT) version of this algorithm (Ke et al. 2017) which decreases the computational time required. The hyperparameters optimised in this case are the depth of the weak learner trees and the number of iterations. The function `HistGradientBoostingClassifier` from `scikit-learn` was used for this algorithm.

3.5 Life sample

The full genomes were collected and analysed by another team member. The data provided to us was a VCF file containing the 69 sets of 20000 SNPs from the four populations (corresponding to the mountain chains of the Alps, the Pyrenees, the Massif Central and the Vosges). The data of the full genomes were analysed in the same way as the simulation data. This gives one datapoint for which we will use the trained machine learning algorithms to predict its probability to belong to one class or another.

4 Results

4.1 General results

We simulated 39445 samples out of the 40000 planned because one step of the analysis failed for unknown reasons. Therefore, we have 9445 samples for Hypothesis02 instead of 10000 but we considered this very light class imbalance as negligible for the following steps.

The data are presented in a PCA and a t-SNE plot (Figure 7). These dimensionality reduction techniques allow us to see that the classification problem at task is probably not linearly separable. This can be shown by the numerous light blue dots (Hypothesis00) within the orange cloud of Hypothesis01 or by the dense mix of datapoints around (0,0) in the PCA. The t-SNE plot also shows that the datapoints are mixed together. The fact that the dataset is nonlinearly separable is also seen by the fact that, despite increasing the number of iterations to 10000, the logistic regression algorithm did not converge.

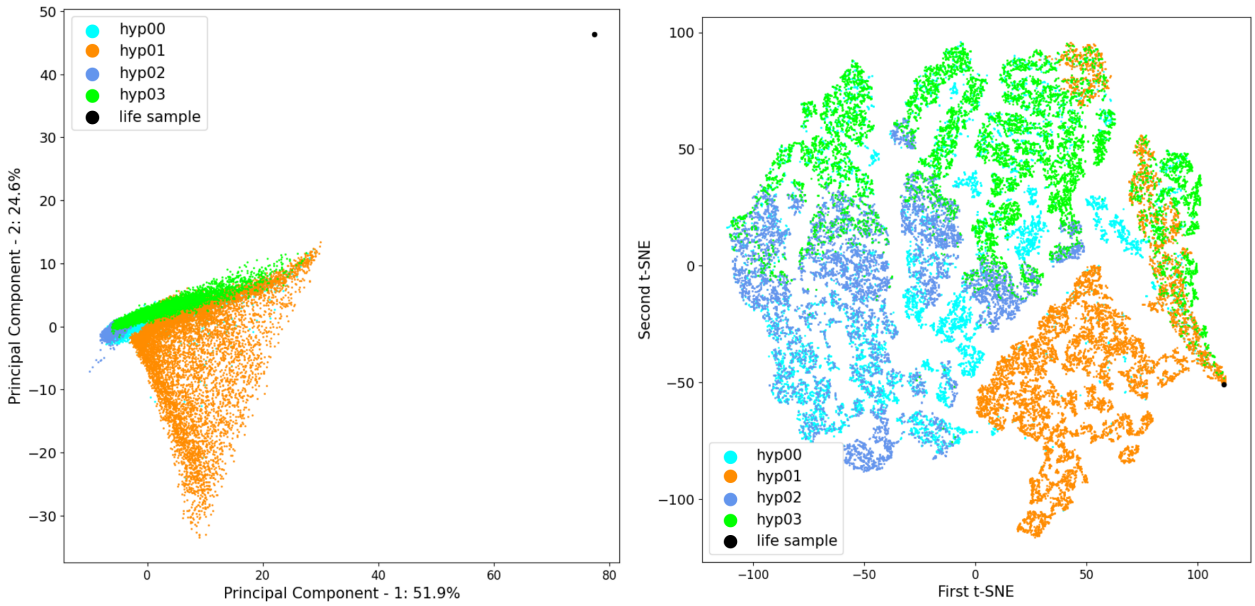

Figure 7: Principal component analysis (PCA) of the entire 39445 simulation data (on the left). The percentage on the axis is the variation explained by those axes. t-SNE of the data (on the right). In both cases, the black datapoint represent the datapoint from the real-life sample.

In terms of computing time, the production of the simulation data took 5h31 on a 4 cores

setting with a laptop CPU. The analyses of the summary statistics were done on the CECI computer cluster and took more than 12 days. This part of the analysis had to be done in parallel on 54 subsets of the data. This high computing time can also be explained by the fact that R does not support multi-threading very well, so we ran the analyses on single CPU cores. Finally, the machine learning pipeline as a whole took 2h32 to run (see details in Table 2). The summary statistics production was thus the clear bottleneck in our pipeline.

Table 2: Computing time required for different steps of the analysis pipeline. The analyses were run on a 16-core laptop.

	Logistic regression	Decision tree	Random Forest	HGBT
First run	40”	15”	1’36”	8”
hyperparameters first run	16’34”	49”	45’56”	18”
hyperparameters second run	27’41”	21”	43’30	2’19”
Validation	7’17”	11”	3’49”	10”
Total	52’	2’	95’	3’
Time per run	33”	0.4”	10”	0.6”

4.2 Accuracy

Given that we have balanced classes, the main metric that we will use to decide which algorithm performs the best is the accuracy. The accuracy of the different classifiers is reported in Table 3 before and after hyperparameters tuning. These results indicate that the two algorithms performing best are the random forest and the HGBT. The hyperparameter tuning did improve the accuracy of the logistic regression and the decision tree but not the one of the random forest and the HGBT.

Table 3: Accuracy score of the different model tested. The accuracy is reported before and after hyperparameter tuning.

	Before hyperparameters tuning	After hyperparameters tuning
Logistic Regression	0.795 ± 0.005	0.824 ± 0.004
Decision Tree	0.784 ± 0.004	0.832 ± 0.007
Random Forest	0.856 ± 0.003	0.855 ± 0.004
HGBT	0.856 ± 0.004	0.856 ± 0.004

To be sure that enough information was brought by the size of the training set we had

decided to use (i.e. 50% of the simulations), we ran the analyses with different fractions of the dataset as training set (i.e. 75% and 99%) and did not see any improvement in the accuracy of the classifiers (not reported here). The following results focus on the 50% split as it gave similar results as the bigger training sets and that it was less computationally heavy to run.

4.3 Confusion matrices

Figure 8 shows the confusion matrices obtained for the four different classification algorithms evaluated in this thesis. We can see that the diagonal is in lighter colour indicating that most of the samples are correctly classified. However, the percentage of well classified samples vary among the hypotheses with Hypothesis01 being the best classified with 100% of correctly classified samples in the case of the random forest.

4.4 ROC curves

Figure 9 shows the different ROC curves for the four algorithms tested in our framework. In the case of a ROC curve, we want to maximise the area under the curve (AUC). This is the case for Hypothesis01 which has an AUC of 1 or close to it for almost all classifiers. We also computed the macro-average ROC curve. This allows for a global estimate of the performance while respecting the class balance. It indicates that all the classifiers perform similarly with a slightly better AUC for the macro-average ROC curve of the HGBT. The ROC curves for Hypothesis02 and Hypothesis03 display similar pattern as the macro-average ROC curve while the ROC curve of Hypothesis00 is less good than average resulting in a lower AUC meaning that this hypothesis is less well predicted than the other three. All the classifiers perform better than random because their ROC curve is above the black dashed diagonal line.

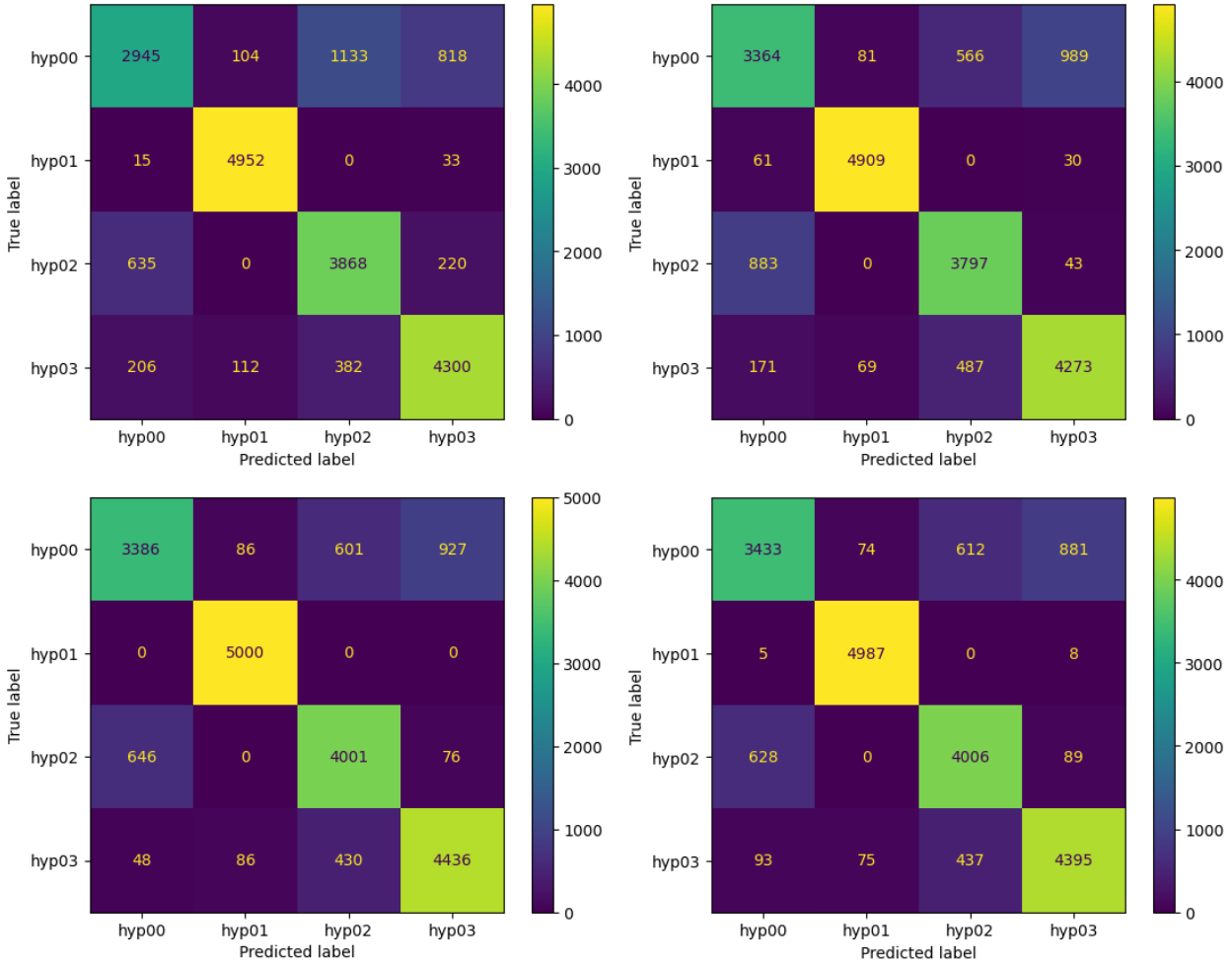

Figure 8: Confusion matrices for the logistic regression (top left), the decision tree (top right), the random forest (bottom left) and histogram-based gradient boosting classification tree (bottom right). The scale ranges from 0 to 5000 and denotes the number of samples from the test set used for a given class. hyp stands for Hypothesis.

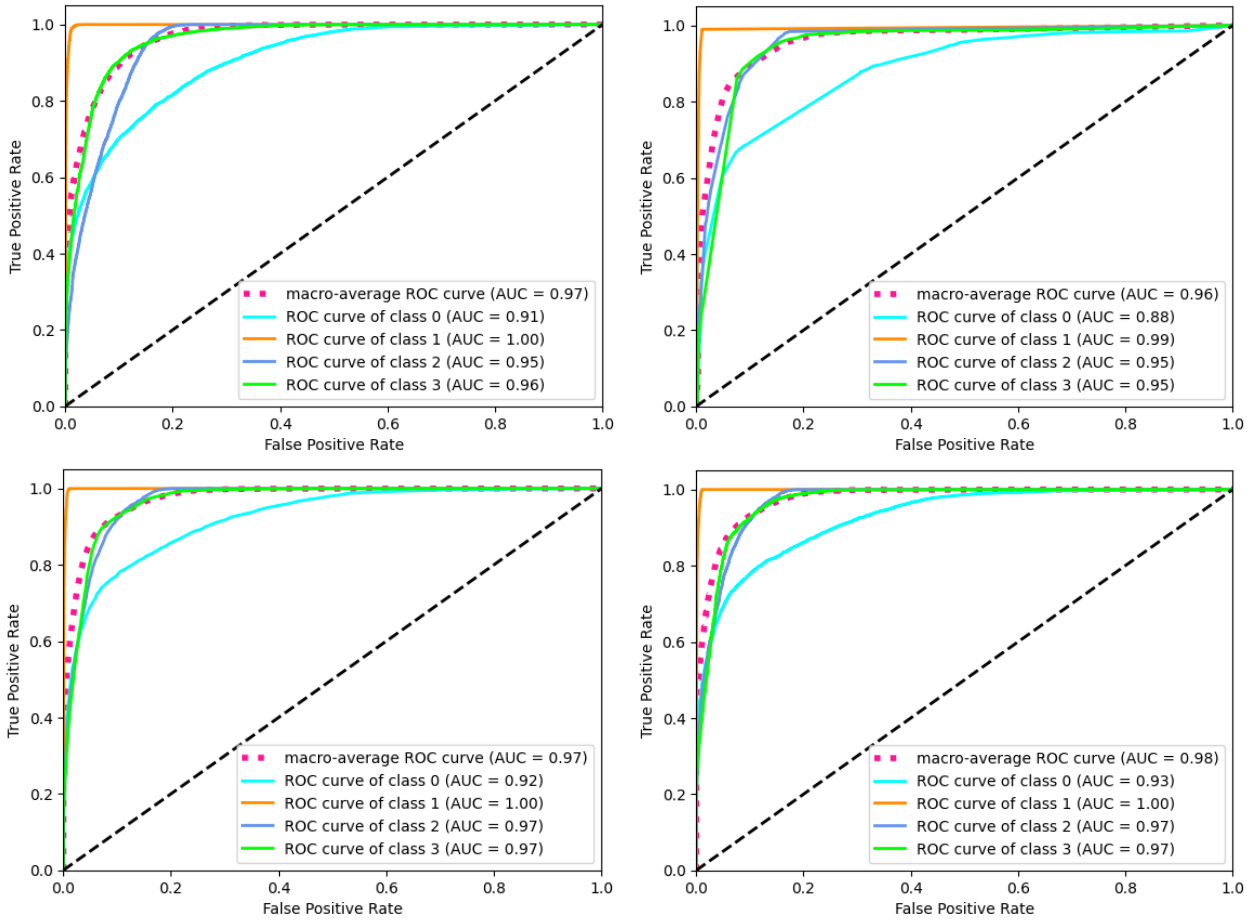

Figure 9: ROC curves for the logistic regression (top left), the decision tree (top right), the random forest (bottom left) and histogram-based gradient boosting classification tree (bottom right). The black dashed diagonal line represents the performance of a dummy classifier. The different class numbers in the legend correspond to hypothesis with the same number.

4.5 Other metrics

Table 4 displays the precision, recall and f1-score for the four classifiers and for each hypothesis. We can observe that the results for the random forest and the HGBT are similar, indicating that they both perform as well on the test set. They both outperform the logistic regression and the decision tree. Overall, the recall is higher than the precision except for Hypothesis00. The f1-score shows that Hypothesis01 is the best predicted by all classifiers.

Table 4: Different classification metrics used for classification. The support column indicates how many samples from the test set were present for each hypothesis.

	precision	recall	f1-score	support
Logistic regression				
hyp00	0.77	0.59	0.67	5000
hyp01	0.96	0.99	0.97	5000
hyp02	0.72	0.82	0.77	4722
hyp03	0.80	0.86	0.83	5000
Decision tree				
hyp00	0.75	0.67	0.71	5000
hyp01	0.97	0.98	0.98	5000
hyp02	0.78	0.80	0.79	4722
hyp03	0.80	0.85	0.83	5000
Random forest				
hyp00	0.83	0.68	0.75	5000
hyp01	0.97	1.00	0.98	5000
hyp02	0.80	0.85	0.82	4722
hyp03	0.82	0.89	0.85	5000
HGBT				
hyp00	0.83	0.69	0.75	5000
hyp01	0.97	1.00	0.98	5000
hyp02	0.79	0.85	0.82	4722
hyp03	0.82	0.88	0.85	5000

4.6 Permutation importance

The permutation importance test allows to rank the features according to their contribution to the models. In our case we ranked the features based on their importance and represented the 20 most important ones in Figure 10. The results show different features being important

for the different classifiers but some trends can be extracted from those data. Across all the algorithms tested here, the following summary statistics seem to come back more frequently: theta, Hs, Hs.1, pi (overall and for individual populations), Fst.1, beta (for different pairs of populations) and dxy (also for different pairs of populations). Their interpretation and meaning will be explored further in the 'Discussion' section.

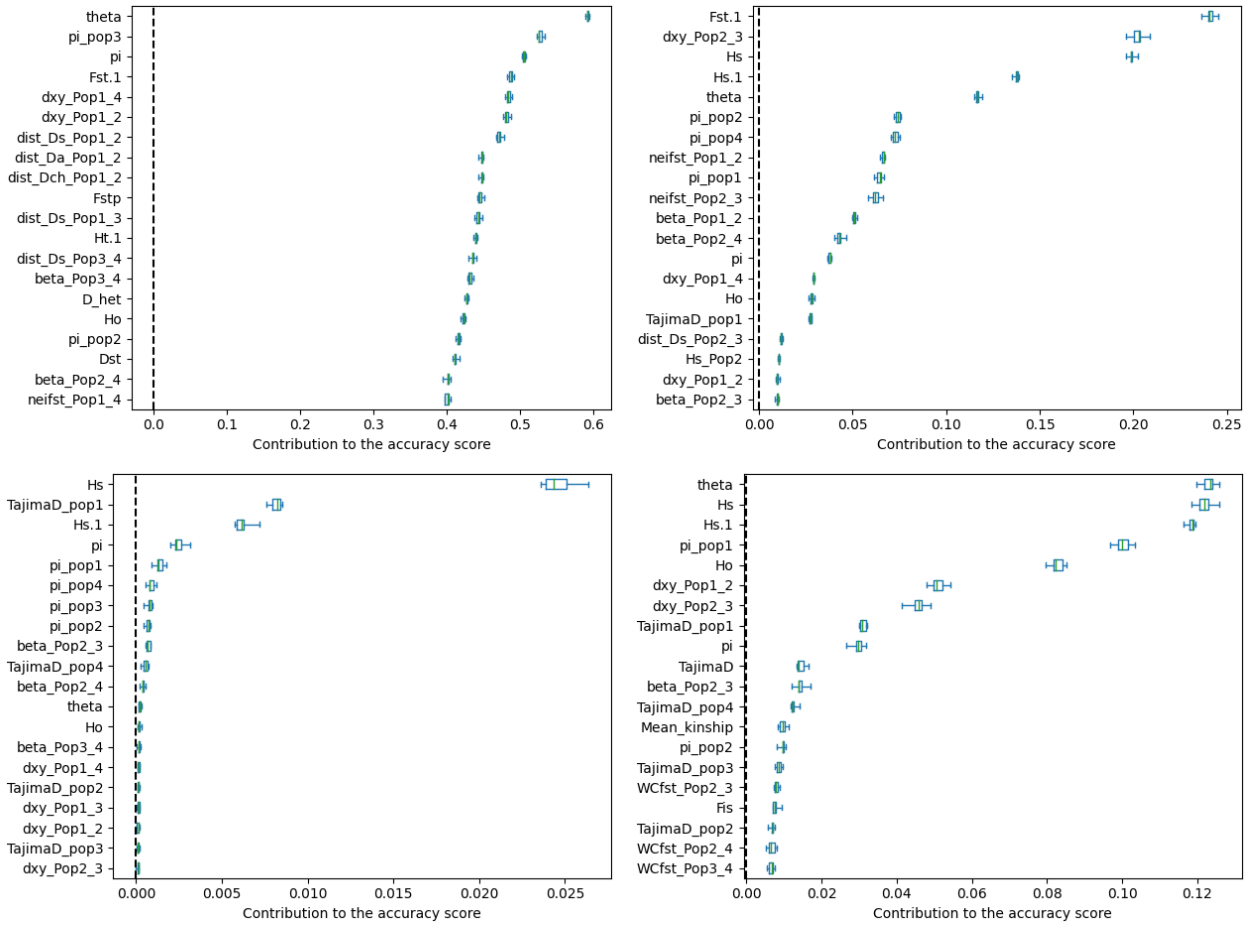

Figure 10: These graphs display the importance of the 20 most important features (i.e. summary statistics) for the logistic regression (top left), decision tree (top right), random forest (bottom left) and for the HGBT (bottom right). The meaning of the different abbreviations can be found in Table 1.

4.7 Life sample

Table 5 references the probabilities of belonging to a certain class for the datapoint of the real-life data from *G quinquepunctata*. Those observed data fall into different categories

depending on the algorithm: the logistic regression predicts the peripheral refuge hypothesis (Hypothesis02), the decision tree and the HGBT favour the lowland refuge hypothesis (Hypothesis01) while the random forest predicts almost equally hypotheses 00, 01 and 03.

Table 5: Results of the predictions for the real-life data per machine learning algorithm.

	Logistic regression	Decision tree	Random forest	HGBT
Hypothesis00	0	0	0.305	0.063
Hypothesis01	0	1	0.324	0.730
Hypothesis02	1	0	0.024	0.003
Hypothesis03	0	0	0.348	0.205

5 Discussion

5.1 Computing time

We observed that the HGBT is 17 times faster than the random forest to analyse our samples. This speed improvement required no trade-off with the accuracy of the classifier nor any of the other metrics evaluated in this thesis. Therefore, it would be relevant to rather use HGBT in such a evolutionary history analysis pipeline.

However, in our pipeline, the main computing time bottleneck was the time taken by the summary statistics production. This could be explained by multiple factors. The first and main one being that most of the packages used were developed to analyse genetic data while our dataset belongs to the genomic era with much more samples to analyse. It impacts particularly the calculation of distance matrices whose size grows greatly with additional samples. Moreover, the distances (dist_ in Table 1) between two populations are not only heavy to compute but also contribute very little to the feature importance (see graphs in Figure 10). Therefore, we could delete these features. The second reason which might explain that this part of the analysis was slow was that R does not support multi-threading natively. We tried to circumvent that problem by massively subsetting the dataset and run analyses on subsets of the data. However, this requires having access to enough CPU cores knowing that we made 54 subsets of the original data.

Another problem encountered which weighed on the pipeline in terms of time and workload was the calculation of the dxy summary statistic. Indeed, there are only two packages known to us capable of calculating this summary statistic: pixy in python and PopGenome in R. We decided to use the latter because pixy was displaying incoherent results in the work of other researcher in the lab. However, at times, the program led to abortion of the R session and we had to rerun the analysis from the moment it stopped. Therefore, it was impossible to run it on the server and we had to run it on a personal laptop. However, this summary statistic showed to contribute to the feature importance of multiple algorithms and was thus necessary to compute. For further research, we would need to get our own implementation of this summary statistic.

5.2 Comparison with other techniques

Compared to traditional ABC methods, using machine learning offers several advantages:

- Only ~20000 samples were used to train our machine learning models instead of hundreds of thousands to millions in traditional ABC method (Pudlo et al. 2016). making the production of simulation data much faster.
- There was no need to find the good trade-off between too few or too many summary statistics, we could select them all.
- HGBT and random forest offer a good interpretability of the probability of belonging to a certain class.
- The permutation importance allows us to see what the contribution of each feature to the classifier is.
- The overall computation time is lower.

However, despite those advantages, there is a major drawback to this approach: the machine learning classifiers will always predict that a given datapoint (like the life sample we had) belongs to a certain class. Therefore, it is important to be careful while describing the evolutionary models that will be used to produce simulation data.

5.3 Most relevant features

The features pointed out to be the most important can be classified into two main categories:

- genetic diversity: Hs and Hs.1 are two different ways of calculating the expected heterozygosity under Hardy-Weinberg equilibrium, it depends on the allele frequency. Theta is characterised by the mutation rate, μ, (fixed in our case) and the effective population size, N_e, which represent the number of individuals actually contributing to the production of the next generation. Indeed, some individuals may be present in the population and not contribute to its genetic dynamic because they do not reproduce. Pi is an indicator of the nucleotide diversity by counting the average number of nucleotide differences between two alleles.

- genetic differentiation: beta is the pairwise kinship coefficient and is one way to measure the similarity between pairs of individuals between populations. Dxy and Fst are similar statistics evaluating the differentiation between populations. However, Fst is calculated based on allele frequencies while the dxy is based on the average number of nucleotide differences per site between populations in a pair of DNA sequences. We can notice that the Fst calculation retained by the algorithms is the one of Nei (1987) and not the one of Weir and Cockerham (1984).

The genetic diversity and differentiation are two approaches to describe genetic data that are complementary to each other, so it makes sense to observe that the machine learning algorithms make use of both type of descriptors.

From most of the permutation importance graphs (i.e. all except the logistic regression one), we can see that few features contribute to the explanation of the model. Therefore, we could have performed a feature selection step which can improve computing time. We did not select features a priori because we wanted to assess the contribution of each features to the model. Tree-based techniques do not suffer from the addition of uninformative features so we decided to keep all features. This simplifies the analyses compared to classical ABC method where it is important to select the right balance of features to not suffer either from the curse of dimensionality or from lack of information.

5.4 Life sample

As we saw previously (see Table 5 in the results), the classification prediction for the real-life data we have are a bit scattered and vary among the algorithm used. We also saw that the data is not linearly separable and that the logistic regression algorithm did not converge. Therefore, we will not consider the prediction result of this algorithm.

Decision trees do not generalise well so we will not focus on the result of this algorithm either but it is still interesting that its prediction fits the one of the HGBT algorithm.

The results obtained by the HGBT seem to favour the lowland refuge hypothesis meaning that, during the last glaciation era, *G quinquepunctata* was one big panmictic populations

and that some groups of individuals colonised the montane area at the end of the glacial era. This hypothesis would fit the spatial distribution modeling as reported by Kastally et al. (submitted).

Finally, the prediction of the random forest is less clear as it gives roughly 33% to three different hypotheses. If we only consider the hypothesis with the highest probability of prediction, the real-life sample would belong to Hypothesis03. This hypothesis, very different from the one predicted by HGBT, states that the colonisation of the Alps would have occurred before the last glacial era and that during this period, there would have been a reduction in population size. At the end of the glacial era, the Pyrenees, the Vosges and the Massif Central would have been colonised by some individuals from the Alps. This result is closer to what Kastally et al. (submitted) had found. However, they used RADseq data and we have more extensive data from the full genomes thus the results are not completely comparable. It is also relevant to remember that in the confusion matrix of the random forest (see Figure 8), 927 samples (i.e. 19%) predicted as Hypothesis03 are in fact belonging to Hypothesis00.

These rather unclear results can be also analysed via the t-SNE graph (see Figure 7) which denotes a proximity between the real-life data and a cluster of points composed of mixed hypotheses (i.e. Hypothesis01 and Hypothesis03).

Finally, it is important to notice the placement of the real-life data in the PCA plot (see Figure 7) far from the other datapoints. This may mean that the hypotheses chosen and simulation data produced cannot encompass the reality of the observed data. One factor that could explain this, is a recent finding of other researchers of the lab, according to which the Alps cannot be considered as one panmictic population. Indeed, the surface area of the Alps is big and could lead to a high gradient of diversity within the mountain chain. Therefore, our initial approximation of considering one population per mountain chain is probably biased and further research should rather consider models which are spatially explicit like the ones included in PhyloGeoSim (Dellicour et al. 2014).

6 Conclusion

To conclude this piece of work, we can say that we developed a pipeline of analysis which performed well to distinguish between the simulation data produced. The classification task is thus possible using supervised machine learning. We propose an alternative to the existing software in the field with different algorithms explored, including HGBT which outperforms the random forest in terms of computing time for this kind of task.

For further research, it would be interesting to turn this pipeline into a software which takes the evolutionary scenario as input and outputs the probability of belonging to a certain evolutionary scenario. It would also be relevant to explore the use of the software `msprime` (Kelleher, Etheridge, and McVean 2016) which is free, open source and written in python. This program is becoming more and more popular and benefits from a growing community of developers participating to the open-source code. Its long-term support and viability are therefore stronger than `fastsimcoal2` that we used to simulate the data. It would also be very important to find an alternative way of computing the summary statistics to decrease the time bottleneck of this part of the pipeline.

For classifying the observed data, a new set of simulation data considering the diversity within the Alps would be necessary.

Acknowledgments

The completion of this work would not have been possible without the help of several people. While there are too many to name them all, I want to thank especially:

- Patrick Mardulyn and Tom Lenaerts, for their presence and insightful comments on my work throughout the year,
- Maeva Sorel and Nassim Versbraegen, for their expertise and guidance,
- Bryan Derbrée, Elisabeth Stryckmans and Martin Neyens, for their support and feedback,
- Carmen Van der Aa, for casting an attentive eye on the grammar and spelling of this thesis.

This research used resources of the "Plateforme Technologique de Calcul Intensif (PTCI)" (http://www.ptci.unamur.be) located at the University of Namur, Belgium, which is supported by the FNRS-FRFC, the Walloon Region, and the University of Namur (Conventions No. 2.5020.11, GEQ U.G006.15, 1610468, RW/GEQ2016 et U.G011.22). The PTCI is member of the "Consortium des Équipements de Calcul Intensif (CÉCI)" (http://www.ceci-hpc.be).

References

Allwein, Erin L, Robert E Schapire, and Yoram Singer. 2000. "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers." *Journal of Machine Learning Research* 1 (Dec): 113–41.

Altmann, André, Laura Toloşi, Oliver Sander, and Thomas Lengauer. 2010. "Permutation Importance: A Corrected Feature Importance Measure." *Bioinformatics* 26 (10): 1340–47.

Aly, Mohamed. 2005. "Survey on Multiclass Classification Methods." *Neural Netw* 19 (1-9): 2.

Amor, A, L Esteve, O Grisel, G Lemaitre, T Schmitt, and G Varoquaux. 2023. "MOOC: Machine Learning in Python with Scikit-Learn."

Anderson, Edgar. 1936. "The Species Problem in Iris." *Annals of the Missouri Botanical Garden* 23 (3): 457–509. http://www.jstor.org/stable/2394164.

Andonie, Răzvan. 2019. "Hyperparameter Optimization in Learning Systems." *Journal of Membrane Computing* 1 (4): 279–91.

Avise, John C. 2000. *Phylogeography: The History and Formation of Species*. Harvard University Press. http://www.jstor.org/stable/j.ctv1nzfgj7.

Beaumont, Mark A. 2010. "Approximate Bayesian Computation in Evolution and Ecology." *Annual Review of Ecology, Evolution, and Systematics* 41: 379–406.

Beaumont, Mark A, Wenyang Zhang, and David J Balding. 2002. "Approximate Bayesian Computation in Population Genetics." *Genetics* 162 (4): 2025–35.

Blum, M. G. B., M. A. Nunes, D. Prangle, and S. A. Sisson. 2013. "A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation." *Statistical Science* 28 (2): 189–208. https://doi.org/10.1214/12-STS406.

Boitard, Simon, Christian Schlötterer, and Andreas Futschik. 2009. "Detecting Selective Sweeps: A New Approach Based on Hidden Markov Models." *Genetics* 181 (March): 1567–78. https://doi.org/10.1534/genetics.108.100032.

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45: 5–32.

Chapuis, Marie-Pierre, Louis Raynal, Christophe Plantamp, Christine N. Meynard, Laurence

Blondin, Jean-Michel Marin, and Arnaud Estoup. 2020. "A Young Age of Subspecific Divergence in the Desert Locust Inferred by ABC Random Forest." *Molecular Ecology* 29 (23): 4542–58. https://doi.org/https://doi.org/10.1111/mec.15663.

Collin, François-david, Ghislain Durif, Louis Raynal, Eric Lombaert, Mathieu Gautier, Renaud Vitalis, Jean-Michel Marin, and Arnaud Estoup. 2021. "Extending Approximate Bayesian Computation with Supervised Machine Learning to Infer Demographic History from Genetic Polymorphisms Using DIYABC Random Forest." *Molecular Ecology Resources* 21 (8): 2598–2613.

Cornuet, Jean-Marie, Pierre Pudlo, Julien Veyssier, Alexandre Dehne-Garcia, Mathieu Gautier, Raphaël Leblois, Jean-Michel Marin, and Arnaud Estoup. 2014. "DIYABC V2. 0: A Software to Make Approximate Bayesian Computation Inferences about Population History Using Single Nucleotide Polymorphism, DNA Sequence and Microsatellite Data." *Bioinformatics* 30 (8): 1187–89.

Cox, David R. 1958. "The Regression Analysis of Binary Sequences." *Journal of the Royal Statistical Society: Series B (Methodological)* 20 (2): 215–32.

Danecek, Petr, Adam Auton, Goncalo Abecasis, Cornelis A Albers, Eric Banks, Mark A DePristo, Robert E Handsaker, et al. 2011. "The Variant Call Format and VCFtools." *Bioinformatics* 27 (15): 2156–58.

Dellicour, Simon, Chedly Kastally, Olivier J Hardy, and Patrick Mardulyn. 2014. "Comparing Phylogeographic Hypotheses by Simulating DNA Sequences Under a Spatially Explicit Model of Coalescence." *Molecular Biology and Evolution* 31 (12): 3359–72.

Excoffier, Laurent, Isabelle Dupanloup, Emilia Huerta-Sanchez, Vitor C Sousa, and Matthieu Foll. 2013. "Robust Demographic Inference from Genomic and SNP Data." *PLoS Genetics* 9 (10): e1003905.

Excoffier, Laurent, and Heidi EL Lischer. 2010. "Arlequin Suite Ver 3.5: A New Series of Programs to Perform Population Genetics Analyses Under Linux and Windows." *Molecular Ecology Resources* 10 (3): 564–67.

Excoffier, Laurent, Nina Marchi, David Alexander Marques, Remi Matthey-Doret, Alexandre Gouy, and Vitor C Sousa. 2021. "Fastsimcoal2: Demographic Inference Under Complex Evolutionary Scenarios." *Bioinformatics* 37 (24): 4882–85.

Fabricius, Johann Christian. 1787. *Ioh. Christ. Fabricii Hist. Nat. Oecon. Et Cameral. P.p.o. Soc. ... Mantissa Insectorum : Sistens Eorum Species Nuper Detectas, Adiectis Characteribus Genericis, Differentiis Specificis, Emendationibus, Observationibus.* Vol. t.1 (1787). Hafniae, Impensis Christ. Gottl. Proft, MDCCLXXXVII [1787]. https://www.biodiversitylibrary.org/item/44030.

Fernández-Delgado, Manuel, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" *The Journal of Machine Learning Research* 15 (1): 3133–81.

Fisher, R. A. 1936. "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics* 7 (2): 179–88. https://doi.org/https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Fraimout, Antoine, Vincent Debat, Simon Fellous, Ruth A Hufbauer, Julien Foucaud, Pierre Pudlo, Jean-Michel Marin, et al. 2017. "Deciphering the Routes of Invasion of Drosophila Suzukii by Means of ABC Random Forest." *Molecular Biology and Evolution* 34 (4): 980–96.

Genuer, Robin, Jean-Michel Poggi, Christine Tuleau-Malot, and Nathalie Villa-Vialaneix. 2017. "Random Forests for Big Data." *Big Data Research* 9: 28–46.

Goudet, Jerome, and Thibaut Jombart. 2015. "Hierfstat: Estimation and Tests of Hierarchical f-Statistics." *R Package Version 0.04-22* 10.

Gruber, Bernd, Peter J Unmack, Oliver F Berry, and Arthur Georges. 2018. "Dartr: An r Package to Facilitate Analysis of SNP Data Generated from Reduced Representation Genome Sequencing." *Molecular Ecology Resources* 18 (3): 691–99.

Gutenkunst, Ryan D. AND Williamson, Ryan N. AND Hernandez. 2009. "Inferring the Joint Demographic History of Multiple Populations from Multidimensional SNP Frequency Data." *PLOS Genetics* 5 (10): 1–11. https://doi.org/10.1371/journal.pgen.1000695.

HackingMaterials. 2019. "Automatminer." https://hackingmaterials.lbl.gov/automatminer/advanced.html.

Hedrick, Philip W. 2005. "A Standardized Genetic Differentiation Measure." *Evolution* 59 (8): 1633–38.

Hoban, Sean, Giorgio Bertorelle, and Oscar E Gaggiotti. 2012. "Computer Simulations:

Tools for Population and Evolutionary Genetics." *Nature Reviews Genetics* 13 (2): 110–22.

Hudson, Richard R. 2001. "Two-Locus Sampling Distributions and Their Application." *Genetics* 159 (4): 1805–17. https://doi.org/10.1093/genetics/159.4.1805.

———. 2002. "Ms a Program for Generating Samples Under Neutral Models." *Bioinformatics* 18 (2): 337–38.

Jombart, Thibaut. 2008. "Adegenet: A r Package for the Multivariate Analysis of Genetic Markers." *Bioinformatics* 24 (11): 1403–5.

Jombart, Thibaut, and Ismail Ahmed. 2011. "Adegenet 1.3-1: New Tools for the Analysis of Genome-Wide SNP Data." *Bioinformatics* 27 (21): 3070–71.

Jost, LOU19238703. 2008. "GST and Its Relatives Do Not Measure Differentiation." *Molecular Ecology* 17 (18): 4015–26.

Kamm, Jack, Jonathan Terhorst, Richard Durbin, and Yun S Song. 2020. "Efficiently Inferring the Demographic History of Many Populations with Allele Count Data." *Journal of the American Statistical Association* 115 (531): 1472–87.

Kastally, Chedly, Maeva Sorel, Flavien Collart, and Patrick Mardulyn. submitted. "Evolution of the Geographic Range of a European Montane Leaf Beetle in Response to Climate Changes at the End of the Quaternary." *Molecular Ecology*, submitted.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree." *Advances in Neural Information Processing Systems* 30.

Kelleher, Jerome, Alison M Etheridge, and Gilean McVean. 2016. "Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes." *PLoS Computational Biology* 12 (5): e1004842.

Kingman, J. F. C. 1982. "The Coalescent." *Stochastic Processes and Their Applications* 13 (3): 235–48. https://doi.org/https://doi.org/10.1016/0304-4149(82)90011-4.

Knaus, Brian J, and Niklaus J Grünwald. 2017. "Vcfr: A Package to Manipulate and Visualize Variant Call Format Data in r." *Molecular Ecology Resources* 17 (1): 44–53.

Larribe, Fabrice, and Paul Fearnhead. 2011. "On Composite Likelihoods in Statistical Genetics." *Statistica Sinica*, 43–69.

Li, Heng. 2011. "A Statistical Framework for SNP Calling, Mutation Discovery, Association Mapping and Population Genetical Parameter Estimation from Sequencing Data." *Bioinformatics* 27 (21): 2987–93.

Li, Na, and Matthew Stephens. 2003. "Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data." *Genetics* 165 (4): 2213–33.

Lischer, Heidi EL, and Laurent Excoffier. 2012. "PGDSpider: An Automated Data Conversion Tool for Connecting Population Genetics and Genomics Programs." *Bioinformatics* 28 (2): 298–99.

Lukicheva, Svitlana, Jean-Francois Flot, and Patrick Mardulyn. 2021. "Genome Assembly of the Cold-Tolerant Leaf Beetle Gonioctena Quinquepunctata, an Important Resource for Studying Its Evolution and Reproductive Barriers Between Species." *Genome Biology and Evolution* 13 (7): evab134.

Marjoram, Paul, John Molitor, Vincent Plagnol, and Simon Tavaré. 2003. "Markov Chain Monte Carlo Without Likelihoods." *Proceedings of the National Academy of Sciences* 100 (26): 15324–28.

Meirmans, Patrick G. 2006. "Using the AMOVA Framework to Estimate a Standardized Genetic Differentiation Measure." *Evolution* 60 (11): 2399–2402.

Mijangos, Jose Luis, Bernd Gruber, Oliver Berry, Carlo Pacioni, and Arthur Georges. 2022. "dartR V2: An Accessible Genetic Analysis Platform for Conservation, Ecology and Agriculture." *Methods in Ecology and Evolution* 13 (10): 2150–58.

Nei, Masatoshi. 1987. *Molecular Evolutionary Genetics*. Columbia university press.

Nei, Masatoshi, and Ronald K Chesser. 1983. "Estimation of Fixation Indices and Gene Diversities." *Annals of Human Genetics* 47 (3): 253–59.

Nei, Masatoshi, and Wen-Hsiung Li. 1979. "Mathematical Model for Studying Genetic Variation in Terms of Restriction Endonucleases." *Proceedings of the National Academy of Sciences* 76 (10): 5269–73.

Nielsen, Rasmus. 2000. "Estimation of Population Parameters and Recombination Rates From Single Nucleotide Polymorphisms." *Genetics* 154 (2): 931–42. https://doi.org/10.1093/genetics/154.2.931.

Nielsen, Rasmus, and Mark A. Beaumont. 2009. "Statistical Inferences in Phylogeography." *Molecular Ecology* 18 (6): 1034–47. https://doi.org/https://doi.org/10.1111/j.1365-294X.2008.04059.x.

Nielsen, Rasmus, Joanna L Mountain, John P Huelsenbeck, and Montgomery Slatkin. 1998. "Maximum-Likelihood Estimation of Population Divergence Times and Population Phylogeny in Models Without Mutation." *Evolution* 52 (3): 669–77.

Nielsen, Rasmus, and Montgomery Slatkin. 2013. *An Introduction to Population Genetics: Theory and Applications*. Sinauer Associates Sunderland, MA.

Pavlidis, Pavlos, Jeffrey D Jensen, and Wolfgang Stephan. 2010. "Searching for Footprints of Positive Selection in Whole-Genome SNP Data From Nonequilibrium Populations." *Genetics* 185 (3): 907–22. https://doi.org/10.1534/genetics.110.116459.

Pearson, Karl. 1901. "On Lines and Planes of Closest Fit to Systems of Points in Space." In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (SIGMOD)*, 19.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Pfeifer, Bastian, Ulrich Wittelsbuerger, Sebastian E. Ramos-Onsins, and Martin J. Lercher. 2014. "PopGenome: An Efficient Swiss Army Knife for Population Genomic Analyses in r." *Molecular Biology and Evolution* 31: 1929–36. https://doi.org/10.1093/molbev/msu136.

Pritchard, Jonathan K, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. 1999. "Population Growth of Human y Chromosomes: A Study of y Chromosome Microsatellites." *Molecular Biology and Evolution* 16 (12): 1791–98.

Pudlo, Pierre, Jean-Michel Marin, Arnaud Estoup, Jean-Marie Cornuet, Mathieu Gautier, and Christian P Robert. 2016. "Reliable ABC Model Choice via Random Forests." *Bioinformatics* 32 (6): 859–66.

Quinlan, J. Ross. 1986. "Induction of Decision Trees." *Machine Learning* 1: 81–106.

Quinzin, Maud C, and Patrick Mardulyn. 2014. "Multi-Locus DNA Sequence Variation in a Complex of Four Leaf Beetle Species with Parapatric Distributions: Mitochondrial

and Nuclear Introgressions Reveal Recent Hybridization." *Molecular Phylogenetics and Evolution* 78: 14–24.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Raynal, Louis, Jean-Michel Marin, Pierre Pudlo, Mathieu Ribatet, Christian P Robert, and Arnaud Estoup. 2019. "ABC Random Forests for Bayesian Parameter Inference." *Bioinformatics* 35 (10): 1720–28.

Rubin, Donald B. 1984. "Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician." *The Annals of Statistics*, 1151–72.

Schapire, Robert E. 1990. "The Strength of Weak Learnability." *Machine Learning* 5: 197–227.

Schmitt, Thomas, Christoph Muster, and Peter Schönswetter. 2010. "Are Disjunct Alpine and Arctic-Alpine Animal and Plant Species in the Western Palearctic Really 'Relics of a Cold Past'?" In *Relict Species: Phylogeography and Conservation Biology*, 239–52. Springer.

Schrider, Daniel R, and Andrew D Kern. 2016. "S/HIC: Robust Identification of Soft and Hard Sweeps Using Machine Learning." *PLoS Genetics* 12 (3): e1005928.

———. 2018. "Supervised Machine Learning for Population Genetics: A New Paradigm." *Trends in Genetics* 34 (4): 301–12.

Scornet, Erwan. 2017. "Tuning Parameters in Random Forests." *ESAIM: Proceedings and Surveys* 60: 144–62.

Sheehan, Yun S., Sara AND Song. 2016. "Deep Learning for Population Genetic Inference." *PLOS Computational Biology* 12 (3): 1–28. https://doi.org/10.1371/journal.pcbi.1004845.

Sigwart, Julia. 2009. "Coalescent Theory: An Introduction." *Systematic Biology* 58 (1): 162–65. https://doi.org/10.1093/schbul/syp004.

Smith, Megan L, and Bryan C Carstens. 2020. "Process-Based Species Delimitation Leads to Identification of More Biologically Relevant Species." *Evolution* 74 (2): 216–29.

Tajima, Fumio. 1989. "Statistical Method for Testing the Neutral Mutation Hypothesis by DNA Polymorphism." *Genetics* 123 (3): 585–95.

Takezaki, Naoko, and Masatoshi Nei. 1996. "Genetic Distances and Reconstruction of Phylogenetic Trees from Microsatellite DNA." *Genetics* 144 (1): 389–99.

Tavaré, Simon, David J Balding, R C Griffiths, and Peter Donnelly. 1997. "Inferring Coalescence Times From DNA Sequence Data." *Genetics* 145 (2): 505–18. https://doi.org/10.1093/genetics/145.2.505.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. "Visualizing Data Using t-SNE." *Journal of Machine Learning Research* 9 (11).

Van Rossum, Guido, and Fred L. Drake. 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Walsh, Ian, Dmytro Fishman, Dario Garcia-Gasulla, Tiina Titma, Gianluca Pollastri, Jennifer Harrow, Fotis E Psomopoulos, and Silvio CE Tosatto. 2021. "DOME: Recommendations for Supervised Machine Learning Validation in Biology." *Nature Methods* 18 (10): 1122–27.

Watterson, GA. 1975. "On the Number of Segregating Sites in Genetical Models Without Recombination." *Theoretical Population Biology* 7 (2): 256–76.

Wegmann, Daniel, Christoph Leuenberger, Samuel Neuenschwander, and Laurent Excoffier. 2010. "ABCtoolbox: A Versatile Toolkit for Approximate Bayesian Computations." *BMC Bioinformatics* 11: 1–7.

Weir, Bruce S, and C Clark Cockerham. 1984. "Estimating f-Statistics for the Analysis of Population Structure." *Evolution*, 1358–70.

Weir, Bruce S, and Jérôme Goudet. 2017. "A Unified Characterization of Population Structure and Relatedness." *Genetics* 206 (4): 2085–2103.

Winter, David J. 2012. "MMOD: An r Library for the Calculation of Population Differentiation Statistics." *Molecular Ecology Resources* 12 (6): 1158–60.

Appendices

```
//Parameters for the coalescence simulation program : fsimcoal2
4 samples to simulate :
//Population effective sizes (number of genes)
ALPES
VOSGES
PYRENEES
MASSIFC
//Samples sizes and samples age
90 //69 in total
10
22
16
//Growth rates  : negative growth implies population expansion
0
0
0
0
//Number of migration matrices : 0 implies no migration between demes
0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix index
5      historical event
Allfrag 1 0 1 1 0 0 //merge to one panmictic population
Allfrag 2 0 1 1 0 0 //merge to one panmictic population
Allfrag 3 0 1 1 0 0 //merge to one panmictic population
Allfrag 0 0 0 Aresize 0 0 //resize the panmictic population
ANCresizeTime 0 0 0 ANCresizeInt 0 0 //allow for coalescence
//Number of independent loci [chromosome]
20000 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block: data type, num loci, rec. rate and mut rate + optional parameters
DNA  10   0   2.1e-7    0.33
```

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #dist.#min  #max
//all Ns are in number of haploid individuals
1  ALPES    logunif  1000   1e4   output
1   Allfrag unif    10000   200000  output
1   TIMEDIFF1   logunif 10  200000  output
0   Aresize logunif 1   1e2 output
0   ANCresizeInt    logunif 1e-4    0.01    output

[RULES]

[COMPLEX PARAMETERS]

1   MASSIFC = 0.25*ALPES    output
1   PYRENEES = 0.1657*ALPES output
1   VOSGES = 0.0335*ALPES   output
1   ANCresizeTime = Allfrag+TIMEDIFF1   output
```

Figure 1: Input files for Hypothesis00 (evolutionary scenario of fragmentation of the populations) with, on the left, the template `.tpl` file and, on the right, the estimation `.est` file.

```
//Parameters for the coalescence simulation program : fsimcoal2
4 samples to simulate :
//Population effective sizes (number of genes)
ALPES
VOSGES
PYRENEES
MASSIFC
//Samples sizes and samples age
90 //69 in total
10
22
16
//Growth rates  : negative growth implies population expansion
0
0
0
0
//Number of migration matrices : 0 implies no migration between demes
0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix index
9     historical event
Allreco 1 1 0 intbotVos 0 0 instbot //pioneer effect aka the end of the glaciation era
Allreco 2 2 0 intbotPyr 0 0 instbot //pioneer effect aka the end of the glaciation era
Allreco 3 3 0 intbotMC 0 0 instbot //pioneer effect aka the end of the glaciation era
Allreco 1 0 1 1 0 0 //back to a single panmictic population
Allreco 2 0 1 1 0 0 //back to a single panmictic population
Allreco 3 0 1 1 0 0 //back to a single panmictic population
Allreco 0 0 0 intbotA 0 0 instbot //pioneer effect aka the end of the glaciation era
Allreco 0 0 0 Aresize 0 0 //increase in size for the peripheral refuge
ANCresizeTime 0 0 0 ANCresizeInt 0 0 //shrinking to allow coalescence
//Number of independent loci [chromosome]
20000 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block: data type, num loci, rec. rate and mut rate + optional parameters
DNA 10 0 2.1e-7 0.33
```

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #dist.#min  #max
//all Ns are in number of haploid individuals
1   ALPES   logunif  1000   1e4   output
0   Aresize logunif 1   1e2 output
1   Allreco unif    10000   12000   output
1   ANCresizeTime   unif    100000  120000  output
0   ANCresizeInt    logunif 1e-4    0.01    output
0   botfactorV  logunif 1e-5    1e-1    output
0   botfactorP  logunif 1e-5    1e-1    output
0   botfactorMC logunif 1e-5    1e-1    output
0   botfactorA  logunif 1e-5    1e-1    output

[RULES]

[COMPLEX PARAMETERS]

1   MASSIFC = 0.25*ALPES    output
1   PYRENEES = 0.1657*ALPES output
1   VOSGES = 0.0335*ALPES   output
1   NbotV = botfactorV*VOSGES output
1   NbotP = botfactorP*PYRENEES output
1   NbotMC = botfactorMC*MASSIFC output
1   NbotA = botfactorA*ALPES output
0   intbotV = 1/NbotV   output
0   intbotP = 1/NbotP   output
0   intbotMC = 1/NbotMC   output
0   intbotA = 1/NbotA   output
```

Figure 2: Input files for Hypothesis01 (evolutionary scenario of lowland refuge) with, on the left, the template `.tpl` file and, on the right, the estimation `.est` file.

```
//Parameters for the coalescence simulation program : fsimcoal2
4 samples to simulate :
//Population effective sizes (number of genes)
ALPES
VOSGES
PYRENEES
MASSIFC
//Samples sizes and samples age
90 //69 in total
10
22
16
//Growth rates  : negative growth implies population expansion
0
0
0
0
//Number of migration matrices : 0 implies no migration between demes
0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix index
13     historical event
Bottlestart 0 0 0 Aresize1 0 0 //size reduction aka end of the glacial era
Bottlestart 1 1 0 Mresize1 0 0 //size reduction aka end of the glacial era
Bottlestart 2 2 0 Presize1 0 0 //size reduction aka end of the glacial era
Bottlestart 3 3 0 Vresize1 0 0 //size reduction aka end of the glacial era
Bottleend 0 0 0 Aresize2 0 0 //size increase aka beginning of the glacial era
Bottleend 1 1 0 Mresize2 0 0 //size increase aka beginning of the glacial era
Bottleend 2 2 0 Presize2 0 0 //size increase aka beginning of the glacial era
Bottleend 3 3 0 Vresize2 0 0 //size increase aka beginning of the glacial era
Timediv 1 0 1 1 0 0 //move to one panmictic population after a while
Timediv 2 0 1 1 0 0 //move to one panmictic population after a while
Timediv 3 0 1 1 0 0 //move to one panmictic population after a while
Timediv 0 0 0 AresizeANC 0 0 //set size of the panmictic population
ANCresizeTime 0 0 0 ANCresizeInt 0 0 //shrinking to allow coalescence
//Number of independent loci [chromosome]
20000 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block: data type, num loci, rec. rate and mut rate + optional parameters
DNA 10 0 2.1e-7 0.33
```

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #dist.#min  #max
//all Ns are in number of haploid individuals
1   ALPES   logunif  1000   1e4   output
0   AresizeANC  logunif 1e-3    100 output
1   Bottlestart unif    10000   12000   output
1   Bottleend   unif    100000  120000  output
1   Timediffrec logunif 10  200000  output
1   Timediffanc logunif 10  200000  output
0   ANCresizeInt    logunif 1e-3    0.01    output
0   Aresize1    logunif 1e-4    1e-1    output
0   Mresize1    logunif 1e-4    1e-1    output
0   Presize1    logunif 1e-4    1e-1    output
0   Vresize1    logunif 1e-4    1e-1    output
0   Aresize2    logunif 1   1e4 output
0   Mresize2    logunif 1   1e4 output
0   Presize2    logunif 1   1e4 output
0   Vresize2    logunif 1   1e4 output

[RULES]

[COMPLEX PARAMETERS]

1   MASSIFC = 0.25*ALPES    output
1   PYRENEES = 0.1657*ALPES output
1   VOSGES = 0.0335*ALPES   output
1   Timediv = Bottleend+Timediffrec output
1   ANCresizeTime = Timediv+Timediffanc output
```

Figure 3: Input files for Hypothesis02 (evolutionary scenario of peripheral refuge) with, on the left, the template `.tpl` file and, on the right, the estimation `.est` file.

```
//Parameters for the coalescence simulation program : fsimcoal2
4 samples to simulate :
//Population effective sizes (number of genes)
ALPES
VOSGES
PYRENEES
MASSIFC
//Samples sizes and samples age
90 //69 in total
10
22
16
//Growth rates  : negative growth implies population expansion
0
0
0
0
//Number of migration matrices : 0 implies no migration between demes
0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix index
12     historical event
Bottlestart 0 0 0 Aresize1 0 0 //shrinking of the population aka end of the glacial era
Bottlestart 1 1 0 Mresize1 0 0 //shrinking of the population aka end of the glacial era
Bottlestart 2 2 0 Presize1 0 0 //shrinking of the population aka end of the glacial era
Bottlestart 3 3 0 Vresize1 0 0 //shrinking of the population aka end of the glacial era
AMCVPSplit 2 2 0 intbotP 0 0 instbot //pioneer effect
AMCVPSplit 2 0 1 1 0 0 //move to the Alps
AMCVPSplit 3 3 0 intbotMC 0 0 instbot //pioneer effect
AMCVPSplit 3 0 1 1 0 0 //move to the Alps
AMCVPSplit 1 1 0 intbotV 0 0 instbot //pioneer effect
AMCVPSplit 1 0 1 1 0 0 //move to the Alps
Bottleend 0 0 0 Aresize2 0 0 //increase in population size at the beginning of the glacial era
ANCresizeTime 0 0 0 ANCresizeInt 0 0 //shrinking to allow coalescence
//Number of independent loci [chromosome]
20000 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block: data type, num loci, rec. rate and mut rate + optional parameters
DNA 10 0 2.1e-7 0.33
```

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #dist.#min  #max
//all Ns are in number of haploid individuals
1   ALPES   logunif  1000   1e4   output
0   AresizeANC  logunif 1e-3    100 output
1   Bottlestart unif    10000   12000   output
1   Bottleend   unif    100000  120000  output
1   AMCVPSplit  unif    11000   110000  output
1   Timediffanc logunif 10  200000  output
0   ANCresizeInt    logunif 1e-3    0.01    output
0   Aresize1    logunif 1e-4    1e-1    output
0   Mresize1    logunif 1e-4    1e-1    output
0   Presize1    logunif 1e-4    1e-1    output
0   Vresize1    logunif 1e-4    1e-1    output
0   Aresize2    logunif 1   1e4 output
0   Mresize2    logunif 1   1e4 output
0   Presize2    logunif 1   1e4 output
0   Vresize2    logunif 1   1e4 output
0   Mresize3    logunif 1   1e4 output
0   botfactorMC logunif 1e-5    1e-2    output
0   botfactorP  logunif 1e-5    1e-2    output
0   botfactorV  logunif 1e-5    1e-2    output

[RULES]

[COMPLEX PARAMETERS]

1   MASSIFC = 0.25*ALPES    output
1   PYRENEES = 0.1657*ALPES output
1   VOSGES = 0.0335*ALPES   output
1   ANCresizeTime = Bottleend+Timediffanc   output
1   NbotMC = botfactorMC*MASSIFC    output
1   NbotP = botfactorP*PYRENEES output
1   NbotV = botfactorV*VOSGES   output
0   intbotMC = 1/NbotMC   output
0   intbotP = 1/NbotP   output
0   intbotV = 1/NbotV   output
```

Figure 4: Input files for Hypothesis03 (evolutionary scenario of peripheral refuge with recolonisation from the Alps) with, on the left, the template `.tpl` file and, on the right, the estimation `.est` file.

DOME	Version	1.0
Data	Provenance	40000 simulation data of four different scenarios (10000 of each) and one datapoint of SNPs extracted from the 69
	Dataset split	Stratified 50%-50% split
	Redundancy between data splits	NA
	Availability of data	Yes for the simulation data on the GitHub page but not yet for the real life SNPs
Optimisation	Algoritm	Logisitc regression
	Meta-predictons	No
	Data encoding	Standard scaling
	Parameters	One, regularisation strength, C
	Algoritm	Decision tree
	Meta-predictons	No
	Data encoding	Standard scaling
	Parameters	One, depth of the tree
	Algoritm	Random forest
	Meta-predictons	No
	Data encoding	Standard scaling
	Parameters	Two, depth of the tree and number of
	Algoritm	Histogram-based gradient boosting classification tree (HGBT)
	Meta-predictons	No
	Data encoding	Standard scaling
	Parameters	Two, depth of the tree and number of
	Features	84
	Fitting	Optimization is a simple majority.
	Regularisation	No
	Availability of configuration	No
Logistic regression	Interpretability	Transparent
	Output	Classification into four classes
	Execution time	52min including the inner cross-validation for hyperparameter tuning
Decision tree	Interpretability	Transparent
	Output	Classification into four classes
	Execution time	2min including the inner cross-validation for hyperparameter tuning
Random forest	Interpretability	Transparent
	Output	Classification into four classes
	Execution time	95min including the inner cross-validation for hyperparameter tuning
HGBT	Interpretability	Transparent
	Output	Classification into four classes
	Execution time	3min including the inner cross-validation for hyperparameter tuning
	Availability of software	On the same GitHub page as the data
Evaluation	Evaluation method	10-fold cross-validation
	Performance measures	Accuracy, precision, recall, f1-score,
	Comparison	Baseline of a dummy classifier
	Confidence	Standard deviation of the cross-validated
	Availability of evaluation	No

Figure 5: Summary of the machine learning procedure used according to DOME guidelines (Walsh et al. 2021).