Invited paper - Special Issue for *Molecular Ecology Resources* on Machine Learning techniques in Evolution and Ecology

Extending Approximate Bayesian Computation with Supervised Machine Learning to infer demographic history from genetic polymorphisms using DIYABC Random Forest

Short title: ABC Random Forest to infer population history

François-David Collin [1], Ghislain Durif [1], Louis Raynal [1], Eric Lombaert [2], Mathieu Gautier [3], Renaud Vitalis [3], Jean-Michel Marin [1,*], Arnaud Estoup [3,*]

[1] IMAG, Univ Montpellier, CNRS, UMR 5149, Montpellier, France

[2] ISA, INRAE, CNRS, Univ Côte d'Azur, Sophia Antipolis, France

[3] CBGP, Univ Montpellier, CIRAD, INRAE, Institut Agro, IRD, Montpellier, France

[*] These authors are joint senior authors on this work

Corresponding author: Arnaud Estoup. E-mail: arnaud.estoup@.inrae.fr

Abstract

Simulation-based methods such as Approximate Bayesian Computation (ABC) are well adapted to the analysis of complex scenarios of populations and species genetic history. In this context, supervised machine learning (SML) methods provide attractive statistical solutions to conduct efficient inferences about scenario choice and parameter estimation. The Random Forest methodology (RF) is a powerful ensemble of SML algorithms used for classification or regression problems. RF allows conducting inferences at a low computational cost, without preliminary selection of the relevant components of the ABC summary statistics, and bypassing the derivation of ABC tolerance levels. We have implemented a set of RF algorithms to process inferences using simulated datasets generated from an extended version of the population genetic simulator implemented in DIYABC v2.1.0. The resulting computer package, named DIYABC Random Forest v1.0, integrates two functionalities into a user-friendly interface: the simulation under custom evolutionary scenarios of different types of molecular data (microsatellites, DNA sequences or SNPs) and RF treatments including statistical tools to evaluate the power and accuracy of inferences. We illustrate the functionalities of DIYABC Random Forest v1.0 for both scenario choice and parameter estimation through the analysis of two example datasets corresponding to pool-sequencing and individual-sequencing SNP datasets. Because of the properties inherent to the implemented RF methods and the large feature vector (including various summary statistics and their linear combinations) available for SNP data, DIYABC Random Forest v1.0 can efficiently contribute to the analysis of large SNP datasets to make inferences about complex population genetic histories.

Key words: Approximate Bayesian Computation, model or scenario selection, parameter estimation, population genetics, Random Forest, Supervised Machine Learning

1 | INTRODUCTION

To keep pace with a regular increase of genetic data accessible to biologists, computational methodologies for population genetic inference are constantly and rapidly being developed. Due to their great flexibility, simulation-based likelihood-free methods such as Approximate Bayesian Computation (ABC; Beaumont, Zhang & Balding, 2002) are well adapted to the analysis of complex models (hereafter referred to as scenarios) of populations and species history, in which divergence events, change of population sizes, and genetic admixture or migration events are suspected (reviewed in Beaumont 2010, Bertorelle, Benazzo, Mona 2010, and Csilléry, Blum, Gaggiotti, & François 2010). With the advent of next generation sequencing (NGS) technologies, population genetic datasets have drastically grown in size (both in terms of number of genotyped loci and number of genetically characterized populations), so that ABC users are facing two major problems: (i) the simulation of massive numbers of large datasets constituting a so called reference table, as required for 'classical' ABC methods, becomes prohibited without extensive computational resources, and (ii) the substantial increase of the number of non-independent statistics used to extract information from the genetic data poses various statistical issues, including the 'curse of dimensionality' whereby accuracy and stability of inferences decrease as the number of summary statistics grows (e.g. Beaumont et al., 2010). Although much effort has gone into dimensionality reduction and feature selection for ABC (reviewed in Blum, Nunes, Prangle, Sisson 2013; Estoup et al., 2012), reducing dimensionality might lead to loss of information if the remaining summaries fail capturing enough information from the data (i.e. if they are not sufficient statistics).

In this context, supervised machine learning (SML) methods provide attractive solutions for statistical inference. SML methods allow predicting new data points through

the use of a training set of labeled simulated data examples, for which true response values are known. This data structure is reminiscent of the ABC reference table. The ability of SML methods to use simulation as a stand-in for observed data is crucial for population genetics applications, where adequately sized datasets with high-confidence labels are currently hard to obtain. Most interestingly, some SML methods are able to take advantage of high dimensional input, suffer only slightly from the curse of dimensionality and are often more robust than other statistical approaches (Chen, Cao, Wen, & Sun, 2013; Anderson, Belkin, Goyal, Rademacher & Voss, 2014; Schrider & Kern, 2018). SML approaches are currently revolutionizing many fields (e.g. Sebastiani 2002 in text categorization; Libbrecht & Noble 2015 in genomics; Angermueller, Parnamaa, Parts, & Stegle 2016 in genomics and cellular imaging), but their use in population genetics inference is still in its infancy (but see e.g. Chapuis et al., 2020; Fraimout et al., 2017; Pybus et al., 2015; Schrider & Kern, 2016, 2017; Sheeman & Song, 2016; Schrider, Ayroles, Matute, & Kern, 2018). The Random Forest (RF) approach proposed by Breiman (2001) is one of the major state-of-the-art SML algorithms for classification (e.g., for scenario choice) or regression (e.g., for estimation of continuous parameters). Pudlo et al. (2016) recently developed RF algorithms to perform scenario choice from simulated datasets summarized through a large set of statistics, as typically considered in ABC, hence leading to the so called ABC-RF approach. As compared to classical ABC methods, the ABC-RF approach enables efficient discrimination among scenarios and estimation of the posterior probability of the best scenario, with a lower computational burden. More specifically, ABC-RF and other ABC methods provide consistent results for analyses based on a large number of simulated datasets, but ABC-RF outperforms other ABC methods for analyses of multiple complex scenarios based on smaller (hence more manageable) number of simulated datasets (Fraimout et al., 2017; Pudlo et al., 2016).

Building on these results, Raynal et al. (2019) recently proposed an extension of the RF approach in a (non-parametric) regression setting to characterize the posterior distributions of parameters of interest in a given scenario. As compared to alternative ABC solutions, Raynal et al.'s (2019) RF method offers many advantages, out of which (i) a significant improvement of robustness to the choice of summary statistics; (ii) the non-requirement of any type of tolerance level; and (iii) a good trade-off between the precision of point estimates of parameters and the accuracy of credible intervals for a given computational burden.

The workflow to applying any SML methods to population genetic data passed through several stages: (i) the simulation of data under one or several evolutionary scenarios; (ii) the encoding of both simulated and real data as feature vectors (i.e., summary statistics as in ABC); and (iii) the training of the algorithm, applying it on new (observed) data point(s), and assessing its performance in term of prediction (through the computation of error and accuracy measurements). Any effort to create self-contained, efficient, and user-friendly software packages capable of performing this entire workflow would streamline SML methods and make them more accessible to researchers, including non-specialist users. To that end, we have implemented in a new computer package a set of RF algorithms to infer population histories from genetic polymorphisms, building upon an extended version of the population genetics simulator implemented in DIYABC 2.1.0 (Cornuet et al., 2014). The data correspond to various types of genetic markers: microsatellites, DNA sequences and SNPs, including pool-sequencing data, which consist of whole-genome sequences obtained from pools of DNA extracted from tens to hundreds of individuals (Gautier et al., 2013; Schlötterer, Tobler, Kofler, & Nolte, 2014). A large set of summary statistics has also been implemented to improve the extraction of genetic information from SNP datasets. The resulting package, named DIYABC Random Forest

v1.0, integrates two functionalities in a user-friendly interface: the simulation under custom evolutionary scenarios of polymorphism data (summarized into a large set of descriptive statistics) and RF treatments including various statistical tools to evaluate the power and accuracy of RF-based inferences. Here we describe the main statistical features of DIYABC Random Forest v1.0 and illustrate its potentialities and functionalities for both scenario choice and parameter estimation through the analyses of two examples of NGS datasets corresponding to pool-sequencing and individual-sequencing SNP data.

2 | METHODS

2.1 | ABC Random Forest in the realm of supervised machine learning

The guiding idea of supervised machine learning (SML) approaches is to use a set of data made of explanatory variables (input) and response values (output), in order to learn the relationship between these two, and hence emit a predicted response value for each new input of interest. More formally, SML methods learn this relationship thanks to a function, f, that predicts a response variable, y, from a feature vector, x, containing M input variables, such that $f(x) = y$. If y is a categorical variable (e.g. for scenario choice), one refers to the task as a classification problem, whereas if y is a continuous variable one refers to it as regression (e.g. for parameter estimation). In supervised learning, the objective is to optimize $f: x \rightarrow y$ using a training set of labeled data (i.e., whose response values are known). The training set includes values of a feature vector which is a multidimensional representation of any data point made up of measurements (or features) taken from it. That is, one assumes to have a set of training data of length n of the form $\{(x_1, y_1),\ldots,(x_n, y_n)\}$, where $x \in \mathrm{R}^M$. A variety of learning algorithms exist which can

generate functions that can perform either classification or regression (reviewed in e.g. Schrider & Kern, 2018)

In our inferential framework, SML methods learn from simulations coming from one or several generative model(s) (i.e. scenario(s)). A relevant way to take benefits from generative scenario simulations is the Bayesian paradigm and therefore the ABC type approach (Beaumont et al., 2002). Here, the training set is equivalent to the ABC reference table, which includes a given number of datasets that have been simulated for different scenarios using parameter values drawn from prior distributions, each dataset being summarized with a set of descriptive statistics. Random Forest (RF; Breiman, 2001) are currently considered as one of the major state-of-the-art SML algorithm for classification or regression. Briefly, RF aggregates the predictions of a collection of classification trees or regression trees, depending on whether the output is categorical (e.g., the identity of a finite number of compared scenarios) or quantitative (e.g., the simulated values of a parameter of interest). Each tree is built by using the information provided by a bootstrap sample of the training set and manages to capture one part of the dependency between the output and the covariates of the feature vector. Based on these random trees which are individually poor to predict the output, a random forest is built by aggregating the tree predictions in order to increase the predictive performances to a high level of accuracy, mainly due to the variance reduction of predictions compared to an individual tree (Breiman, 2001). More details and in-depth explanations can be found in Pudlo et al. (2016), Fraimout et al. (2017), Estoup, Raynal, Verdu, & Marin (2018) and Marin, Pudlo, Estoup, & Robert (2018) for scenario choice, and Raynal et al. (2019) for parameter estimation. See also the Supplementary Material S3 of Chapuis et al. (2020) for a concise overview of the RF algorithms and statistical developments used in the present paper and implemented in the computer package DIYABC Random Forest v1.0.

2.2 | Simulation of the training set

Before processing RF analyses, one needs to generate a training set. The datasets composing the training set can be simulated under different scenarios and sample configurations, using parameter values drawn from prior distributions. Each resulting dataset is summarized using a set of descriptive statistics. We formalized scenarios and prior distributions, and computed summary statistics using the "Training set simulation" module of DIYABC Random Forest v1.0, which essentially corresponds to an extended version of the population genetics simulator implemented in DIYABC v2.1.0 (Cornuet et al., 2014). As in the latter program, DIYABC Random Forest v1.0 allows considering complex population histories including any combination of population divergence events, symmetrical or asymmetrical admixture events (but not any continuous gene flow between populations) and changes in past population size, with population samples potentially collected at different times.

DIYABC Random Forest v1.0 accepts various types of molecular data (microsatellites, DNA sequences, and SNPs) evolving under various mutation models and located on various chromosome types (autosomal, X or Y chromosomes, and mitochondrial DNA) for diploid or haploid individuals. Compared to other types of markers, SNP loci have low mutation rates, so that polymorphism at such loci is assumed to be caused by a single mutation that occurred along the whole population(s) gene tree, which results in biallelic genotypes. To simulate polymorphic datasets at a given SNP locus, we follow the algorithm proposed by Hudson (1993) – cf. *–s 1* option in the program ms associated to Hudson (2002) – that consists in fixing one segregating site in the genealogy and thus leads to applying a default MAF (minimum allele frequency) criterion on the simulated dataset. As a matter of fact, each locus in both the observed and simulated

datasets will be characterized by the presence of at least one copy of the SNP alleles over all genes sampled from all studied populations (i.e. pooling all genes genotyped at the locus). In DIYABC Random Forest v1.0, it is possible to impose a different MAF criterion for each locus on the observed and simulated datasets. This MAF is computed pooling all genes genotyped over all studied population samples. For instance, the specification of a MAF equal to 5% will automatically select a subset of *m* loci characterized by a minimum allele frequency > 5% out of the *l* loci of the observed dataset. In agreement with this, only *m* loci with a MAF>5% will be retained in a simulated dataset.

In addition to individual-sequencing SNP data (hereafter IndSeq data), DIYABC Random Forest v1.0 allows the simulation and analyses of pool-sequencing SNP data (hereafter PoolSeq data), which basically consist of whole-genome sequences of pools of tens to hundreds of individual DNAs (Gautier et al., 2013; Schlötterer et al., 2014). In practice, the simulation of PoolSeq data consists first in simulating individual SNP genotypes for all individuals in each population pool, and then generating pool read counts from a binomial distribution parameterized with the simulated allele counts (obtained from individual SNP genotypes) and the total pool read coverage (e.g., Hivert, Leblois, Petit, Gautier, & Vitalis, 2018). To account for variation of the total read coverage across SNPs as observed in the actual dataset, the coverages across the pools of a given SNP are randomly drawn from the vectors of SNP coverages composing the observed dataset. The "Synthetic data file generation" module of the program allows the simulation of various types of pseudo-observed 'raw' datasets (i.e. not summarized through statistics) without referring to any (actual) observed dataset. In the case of raw PoolSeq datasets, the total coverage within each pool of each SNP is sampled from a Poisson distribution with a mean corresponding to an arbitrary coverage value (e.g. 100X) fixed by the DIYABC Random Forest v1.0 user.

It is worth noting that, in contrast to any other types of markers considered by DIYABC Random Forest v1.0 (including IndSeq SNPs), PoolSeq SNP data are considered as located on autosomal chromosomes only. A criterion somewhat similar to the MAF was implemented for PoolSeq data: the minimum read count (MRC). The MRC is the number of sequence reads of the minor allele frequency allele when pooling the reads over all population samples. The specification of a MRC equal for instance to 5 will automatically select a subset of *m* PoolSeq loci characterized by more than five reads over all studied pools among the *l* loci of the observed dataset. In agreement with this, only *m* loci with more than five reads will be retained in a simulated dataset.

2.3 | Components of the feature vector

The feature vector includes a large number of statistics that summarizes genetic variation in the way that they allow capturing different aspects of gene genealogies and hence various features of molecular patterns generated by selectively neutral population histories (e.g. Beaumont 2010; Cornuet et al., 2014). For microsatellite and DNA sequence markers, DIYABC Random Forest v1.0 proposes by default the same set of summary statistics as DIYABC v2.1.0 (Cornuet et al., 2014). These summary statistics describe genetic variation within population (e.g. numbers of alleles), between pair (e.g., genetic distances), or per triplet (e.g., coefficients of admixture) of populations, averaged over loci; see details in the DIYABC Random Forest v1.0 user manual (https://diyabc.github.io).

For both IndSeq and PoolSeq SNPs, we have implemented in DIYABC Random Forest v1.0 an extended set (when compared to DIYABC v2.1.0) of summary statistics. The proportion of monomorphic loci is computed for each population, as well as for each pair and triplet of populations. Mean and variance (over loci) values are computed for all subsequent summary statistics. Heterozygosity is computed for each population and for

each pair of populations as (1 - Q_1) and (1 – Q_2), where Q_1 and Q_2 are the probabilities of identity between pairs of genes described in the Supplementary File S1 of Hivert et al. (2018). F_{ST}-related statistics are computed for each population (i.e., population-specific F_{ST} as described in Weir & Goudet 2017), as well as for each pair, triplet, quadruplet and overall populations (when the dataset includes more than four populations), using the method-of-moments estimators described in Hivert et al. (2018). In addition, we compute Patterson's *f*-statistics for each triplet (f_3-statistics) and quadruplet (f_4-statistics) of populations as described in Patterson et al. (2012), except for the f_3-statistics for PoolSeq read count data which are computed using the unbiased estimator described in Leblois et al. (2018). Finally, Nei's (1972) distance is computed for each pair of populations and the coefficient of admixture is computed for each triplet of populations as described in Cornuet et al. (2014). For additional details, see the user manual of DIYABC Random Forest v1.0 (https://diyabc.github.io). An illustration of the feature vector composed of all above summary statistics is given in Table S1 for the analysis of two example SNP pseudo-observed datasets.

For scenario choice, the feature vector can be enriched by values of the *d* axes of a linear discriminant analysis (LDA) processed on the above summary statistics (with *d* equal to the number of scenarios minus 1; Pudlo et al., 2016). In the same spirit, for parameter estimation, the feature vector can be completed by values of a subset of the *s* axes of a Partial Least Squares Regression analysis (PLS) also processed on the above summary statistics (with *s* equal to the number of summary statistics). The number of PLS axes added to the feature vector is determined as the number of PLS axes providing a given fraction of the maximum amount of variance explained by all PLS axes (i.e., 95% by default, but this parameter can be adjusted).

2.4 | Prediction using Random Forest

We used the "Random Forest analyses" module of the software DIYABC Random Forest v1.0 to process RF prediction applied to a given target dataset. For scenario choice, the outcome of the first step of RF computation is a classification vote for each scenario which represents the number of times a scenario is selected in a forest of *n* trees. The scenario with the highest classification vote corresponds to the scenario best suited to the target dataset among the set of compared scenarios. This first RF predictor is good enough to select the most likely scenario but not to derive directly the associated posterior probabilities. A second analytical step based on a second Random Forest in regression is necessary to provide an estimation of the posterior probability of the best supported scenario (Pudlo et al., 2016). For parameter estimation, Raynal et al. (2019) extended the RF approach developed in the context of (non-parametric) regression (Breiman, 2001), to estimate the posterior distributions of parameters of interest in a given scenario. The approach requires the derivation of a new Random Forest for each component of interest of the parameter vector. Quite often, practitioners of Bayesian inference report the posterior mean, posterior variance or posterior quantiles, rather than the full posterior distribution, since the former are easier to interpret than the latter. We implemented the methodologies detailed in Raynal et al. (2019) to provide estimations of the posterior mean, variance, median (i.e. 50% quantile) as well as 5% and 95% quantiles (and hence 90% credibility interval) of each parameter of interest. The posterior distribution of each parameter of interest is obtained using importance weights following Meinshausen (2006)'s work on quantile regression forests.

2.5 | Assessing the quality of predictions

For scenario choice and parameter estimation, DIYABC Random Forest v1.0 allows evaluating the robustness of inferences. Because the level of errors on scenario choice and accuracy of parameter estimation may substantially differ depending on the location of an observed dataset in the prior data space, prior-based indicators are poorly relevant, aside from their use to select the best classification method and possibly a set of highly informative components of the feature vector. Therefore, in addition to global (i.e. prior) error/accuracy corresponding to prediction quality measures computed over the entire data space, it is crucial to compute local (i.e. posterior) error/accuracy conditionally on the observed dataset, corresponding to prediction quality exactly at the position of the observed dataset. For scenario choice, the global prior errors including the confusion matrix (i.e. the contingency table of the true and predicted classes for each example in the training set) and the mean misclassification error rate were computed using the out-of-bag (a.k.a. out-of-bootstrap) training data as free test dataset. The out-of-bag dataset corresponds to the data of the training set that were not selected when creating the different tree bootstrap samples (Breiman, 2001). For scenario choice, Chapuis et al. (2020) highlighted that the local (posterior) error can be computed as 1 minus the posterior probability of the selected scenario. For parameter estimation, we also relied on out-of-bag predictions to compute both global (i.e. prior) and local (i.e. posterior) accuracy measures, as detailed in the Supplementary Material S3 of Chapuis et al. 2020 (see also Raynal et al. 2019). DIYABC Random Forest v1.0 includes the following accuracy measures: (i) both the global and local NMAE (i.e., the normalized mean absolute error which is the average absolute difference between the point estimate and the true simulated value divided by the true simulated value) with the mean or the median taken as point estimate; ii) both the global and local MSE and NMSE (i.e., the mean square error which is the average squared difference between the point estimate and the true simulated value for MSE, divided by the

true simulated value for NMSE), again with the mean or the median taken as point estimate; and iii) several confidence interval measures, computed only at the global scale, including the 90% coverage (i.e., the proportion of true simulated values located between the estimated 5% and 95% quantiles), and the mean or the median of the 90% amplitude and relative 90% amplitude (i.e., the mean or median of the difference between the estimated 5% and 95% quantiles for the 90% amplitude, divided by the true simulated value for the relative 90% amplitude).

It is worth stressing that using the out-of-bag prediction method for estimating global and local error/accuracy measures is computationally efficient as this approach makes use of the datasets already present in the training set and hence avoids the computationally costly simulations (especially for large SNP datasets) of additional test datasets.

2.6 | Implementation

The package DIYABC Random Forest v1.0 is composed of three parts: the dataset simulator, the Random Forest inference engine and the graphical user interface. The whole is packaged as a standalone and user-friendly application available at https://diyabc.github.io. The different developer and user manuals for each component of the package are available on the same site. DIYABC Random Forest v1.0 is a multithreaded program which runs on three operating systems: GNU/Linux, Microsoft Windows and MacOS. Computational procedures of the simulator and the Random Forest inference engine are written in C++. The graphical user interface is written in R shiny (Chang, Cheng Allaire, Xie, & McPherson, 2019) and available as a standalone graphical application or as a R package implementing a web application that can be run locally or hosted as a web service.

For the Random Forest part of DIYABC Random Forest v1.0, we used our own version of the core RF (written in C++) from the package ranger (Wright & Ziegler 2017). In this new version, that we named abcranger, the Random Forest computations are optimized in order to grow a limited batch of trees in memory (but still computed in parallel to leverage multicore architectures) in sequential – i.e. batch-wise order. Tree growing and predictions are computed in a single pass, predictions are stored or accumulated and each tree is then discarded. Although we still need the entire training set at once, processing in this way avoids the in-memory storage of the whole forest at zero performance cost. The abcranger package hence opens new perspective to efficiently compute RF from training sets of (very) large size. For instance, a training set including > 100,000 particles of a feature vectors composed of > 10,000 summary statistics could be treated without any memory overflow (results not shown). It is worth stressing that abcranger is not limited to population genetics applications as the program can be used as an inference engine independently from the DIYABC simulator. However, for the moment, the binary standalone used by the DIYABC interface handles only outputs produced by the DIYABC simulator. A python wrapper (and example notebooks) is available at https://github.com/diyabc/abcranger and a R wrapper will be soon provided at the same site.

2.7 | Interface and outputs

DIYABC Random Forest v1.0 can be used through a modern and user-friendly graphical interface designed as an R shiny application (Chang et al., 2019). For a fluid and simplified user experience, this interface is available through a standalone application, which does not depend on R and hence can be used independently. The application is also implemented in a R package providing a standard shiny web application (with the same graphical interface)

that can be run locally as any shiny application, or hosted as a web service to provide a DIYABC Random Forest v1.0 server for multiple users.

The interface is divided into two modules corresponding to the two main phases of a statistical treatment based on DIYABC Random Forest v1.0: module 1 = “Training set simulation” and module 2 = “Random Forest analyses”. In module 1, users specify what type and how simulated data will be generated under the ABC framework to produce a training set. Module 2 guides users through scenario choice and parameter inference by providing a simple interface for the supervised learning framework based on Random Forest methodologies. An additional module named “Synthetic data file generation” (based on the DIYABC simulation engine) is also available in the application. It can be used to easily generate datafile(s) for various types of genetic markers corresponding to synthetic “ground truth” raw data (not summarized through statistics) under a given historical scenario and a set of fixed parameter values. The formats of the generated datafiles are similar to those of the observed input datafiles read by DIYABC Random Forest v1.0 (for details see user manual at https://diyabc.github.io).

The integration of the various graphical outputs (historical scenario representation, error or accuracy indices, posterior curves, contribution to inferences of components of the feature vector, etc.) is managed with the ggplot2 R package (Wickham 2016), allowing user to create and export high-quality graphics related to the analyses. We encourage users to consult the user manual of the program available at https://diyabc.github.io for details regarding the various numerical and graphical outputs provided by DIYABC Random Forest v1.0. It is worth noting that a number of such outputs have been used in the present paper to illustrate the results obtained when analyzing two example SNP pseudo-observed datasets (see below).

2.8 | Illustration using two example pseudo-observed SNP datasets

Compared scenarios and prior distributions

We considered a case study where one wants to make inferences about the genetic origin of a population of interest (for example a recent invasive population) among a set of possible source populations (for which the topology is known; see Figure 1). The target population (pop 4) has three possible single population sources (pop1, pop2 or pop3) and three possible admixed pairwise population sources (i.e., admixture between pop1 & pop2, pop1 & pop3 and pop2 & pop3). We hence formalized six competing scenarios that constitute two groups of scenarios when referring to the presence or absence of an admixture event when founding the target population 4: group 1 includes three scenarios including an admixture event (scenarios 1, 2 and 3) and group 2 three scenarios without any admixture event (scenarios 4, 5 and 6) (Figure 1). Such grouping approach in scenario choice is relevant to disentangle in our analysis the level of confidence to make inferences about a given (or several) specific evolutionary event of interest, here the presence or absence of an admixed origin of population 4 (Estoup et al., 2018; Chapuis et al., 2020).

Demographic and historical parameters include four effective population sizes N_1, N_2, N_3 and N_4 (for populations 1, 2, 3, and 4, respectively) and three divergence or admixture time events (t_1, t_2 and t_3), with t_1 the divergence or admixture time of pop4, t_2 the divergence time of pop3 from pop2, and t_3 the divergence time of pop2 from pop1 (Figure 1). For the three scenarios with admixture, the parameter r_a corresponds to the proportion of genes of a given source population entering into the admixed population 4, as described in Figure 1. Prior values for time events (t_1, t_2, and t_3) were drawn from uniform distributions bounded between 10 and 1,000 generations, with $t_3 > t_2 > t_1$. We used uniform prior distributions bounded between $1x10^2$ and $1x10^4$ diploid individuals for each

423 effective population sizes N_1, N_2, N_3 and N_4. The admixture rate r_a was drawn from a
424 uniform prior distribution bounded between 0.05 and 0.95.
425
426 *Pseudo-observed datasets*
427 Our prediction targets correspond to two pseudo-observed datasets that were generated
428 using the "Synthetic data file generation" module of DIYABC Random Forest v1.0 under
429 the scenario 3 using the following parameter values: N_1=7000, N_2=2000, N_3=4000,
430 N_4=3000, t_1=200, r_a=0.3, t_2=300 and t_3=500. The short divergence times and large
431 effective population sizes values corresponds to a situation of low level of genetic
432 differentiation among populations (cf. parwise F_{ST} values ranging from 3% to 7%) and
433 hence to a difficult case study. The two pseudo-observed datasets correspond to a PoolSeq
434 read count dataset and an IndSeq allele count dataset, each with 30,000 SNPs. They
435 represent similar sequencing efforts: a 100X coverage for each population of the PoolSeq
436 dataset (with 100 individuals per population pool) and 10 individuals sequenced per
437 population for the IndSeq dataset with a 10X coverage for each sequenced individual (the
438 latter parameter being not explicitly indicated in the program as individual SNP genotypes
439 are considered to be inferred without errors). RF analyses were processed with DIYABC
440 Random Forest v1.0 on a subset of 5,000 SNPs with a MRC = 5 for the PoolSeq dataset
441 and a MAF = 5% for the IndSeq dataset.
442
443 *Scenario choice*
444 Following the new approach proposed by Estoup et al. (2018), we used DIYABC Random
445 Forest v1.0 to process RF analyses grouping scenarios based on the presence or absence of
446 an admixed origin of population 4, and then considered all six scenarios separately. The
447 training set were generated using the "Training set simulation" module of DIYABC

Random Forest v1.0, drawing parameter values into the prior distributions described above and summarizing SNP data using 130 statistics (see Table S1 for details about such summary statistics) plus one LDA axis or five LDA axes (i.e., the number of scenarios minus 1; see Pudlo et al. 2016) computed when comparing the two groups of scenarios or individual scenarios, respectively. We then used the "Random Forest analyses" module of DIYABC Random Forest v1.0 to process RF treatments on the training sets which included a total of 12,000 simulated datasets (i.e., 2,000 per scenario). Following Pudlo et al. (2016), we checked that 12,000 datasets in the training set was sufficient by evaluating the stability of prior error rates and posterior probabilities estimations of the best scenario on subsets of 10,000, 11,000 and 12,000 data of the training set (results not shown). The number of trees in the constructed Random Forest was fixed to 1,000, as this number turned out to be large enough to ensure a stable estimation of the global error rate (Figure S1). We predicted the best scenario and estimated its posterior probability, as well as the global and local error rates, over ten replicate RF analyses based on the same training set.

Parameter estimation

Following Raynal et al. 2019, we conducted independent RF treatments for each parameter of interest. For the sake of concision, we focused our estimations on four parameters involved in the admixture event in scenario 3 (i.e. the selected scenario after processing scenario choice): the founding/admixture time for the target population 4 (t_1), the admixture rate (r_a corresponding to the proportion of genes originating from population 1), the effective population size of population 4 (N_4), and the compound parameter corresponding to the ratio t_1/N_4. As a matter of fact, considering ratios (or products) of parameters - here the admixture time scaled by the effective population size as drift parameter - allows reducing parameter identifiability issues of some scenarios (e.g.,

Beaumont 2010). The training sets included 10,000 datasets simulated under scenario 3 and summarized using the same 130 statistics (Table S1) plus 4 to 24 PLS axes depending on the parameter estimated and the training set analyzed. For each parameter, we inferred point estimates and computed global and local accuracy indices corresponding to global and local NMAE using out-of-bag estimators from a sample of 10,000 data randomly chosen in the training set, with the mean and the median as point estimates). We checked that 10,000 datasets in the training set were sufficient by evaluating the stability of the global accuracy indices (i.e., NMAE using the mean as point estimates) on subsets of 8,000, 9,000 and 10,000 data of the training set (results not shown). The number of trees in the constructed Random Forest was fixed to 1,000, as this number turned out to be large enough to ensure a stable estimation of the global accuracy metrics (Figure S1). For each parameter, we conducted ten replicate RF analyses based on the same training set.

Computing time and memory space

All analyses on the example pseudo-observed datasets were processed on a 16 cores Intel Xeon E5-2650 computer (Linux Debian platform, 64 bits system), with a maximum of 26 Gb and 1.8 Gb of RAM used for the heaviest treatments regarding the simulation of the training set (with a loop-size of 50 datasets corresponding to the number of simulated datasets distributed over all computer threads) and RF analyses, respectively. Optimizing computer code procedures to efficiently compute summary statistics is important especially in the case of high-dimensional analyses which may include several thousand summary statistics. Substantial efforts in this direction on DIYABC Random Forest v1.0 allowed to considerably reduce (compared to the simulation module of DIYABC v2.1.0) both the fraction of the running time and the memory space devoted to the computation of summary statistics. Such optimizations open new perspectives for the analysis of (very) high-

dimensional datasets in population genetics. The production of a training set including 10,000 simulated datasets took 13 min (respectively 26 h) with only 4% (respectively 10%) of the running time devoted to the computation of the 130 summary statistics for the IndSeq (respectively PoolSeq) data. Note that the computation time difference between IndSeq and PoolSeq reflects the ten time larger number of individuals included in the PoolSeq simulation setting. RF treatments following the generation of the training set took less than 30 sec for scenario choice and 1 min for each parameter estimation (with 37% of the time used to compute local NMAE accuracy measures estimated using out-of-bag estimators from a sample of 10,000 data randomly chosen in the training set).

3 | RESULTS

For both scenario choice and parameter estimation, we illustrate the inferential power and functionalities of DIYABC Random Forest v1.0 through the analysis of two example pseudo-observed SNP datasets corresponding to PoolSeq and IndSeq data. We first processed RF analyses grouping scenarios based on the presence or absence of an admixed origin of the target population 4, and then considered all six compared scenarios separately. We then estimated parameters of interests under the selected (best) scenario. We contrasted our inferential results with and without adding LDA axes (for scenario choice) or PLS axes (for parameter estimation) to the RF feature vector initially composed of 130 summary statistics (Table S1).

3.1 | Scenario choice

The projection of the datasets of the training set on a single (when analyzing the two groups of scenarios) or on the first two LDA axes (when analyzing the six scenarios

523 considered separately) provides a first visual indication about our capacity to discriminate
524 among the compared scenarios (Figure 2). Simulations under the two groups of scenarios
525 moderately overlapped suggesting a substantial power to discriminate among them. When
526 considering the six scenarios individually, the projected points overlapped in a more
527 marked way, at least for some of the scenarios, suggesting an overall lower power to
528 discriminate among scenarios considered separately than when considering the two groups
529 of scenarios. As a first inferential clue, the location of the observed dataset (indicated by a
530 vertical line and a star symbol in Figure 2A and 2B, respectively) suggests, albeit without
531 any formal quantification, a marked association with the scenario group 1 and with the
532 scenario 3.

533 The classification votes and posterior probabilities estimated for both example
534 pseudo-observed datasets (with or without adding LDA axes to the feature vector) were the
535 highest for the scenario group 1, which includes an admixture event (Table 1). When
536 considering the six scenarios separately, the highest classification votes and posterior
537 probabilities were for scenario 3, which congruently includes a genetic admixture event
538 between the population 1 and 3 as sources of the target population 4. The posterior
539 probabilities of scenario group 1 and scenario 3 were relatively high (from 0.657 to 0.891
540 depending on the analysis), which is satisfactory when considering the difficulty of the
541 example case study (cf. low level of genetic differentiation among populations). We found
542 that including LDA axes in the RF vector feature substantially improved scenario choice
543 predictions. Global (prior) error rates were 3% to 12% lower when including LDA axes.
544 Regarding our two pseudo-observed datasets, classification votes for the best group were
545 4% to 8% higher when including LDA axes and posterior probabilities of the selected
546 scenario 3 (which is equal to 1 minus the local error rate) were 8% to 12% higher when
547 including LDA axes. The levels of errors were substantially different at the global and

548 local scales, with lower levels at the local scale for analyses of the PoolSeq dataset, and a
549 trend for higher levels at the local scale for analyses of the IndSeq dataset.
550 Finally, we obtained better prediction levels (with or without LDA axes) for the
551 PoolSeq dataset than the IndSeq dataset. Global (prior) error rates were 14% to 27% lower
552 for the PoolSeq dataset. Regarding our two pseudo-observed datasets, classification votes
553 for the best group were 8% to 11% higher for the PoolSeq dataset and posterior
554 probabilities of the selected scenario 3 were 11% to 21% higher for the PoolSeq dataset.
555 This indicates that, for a similar sequencing effort, a PoolSeq strategy is preferable to an
556 IndSeq strategy, at least when a substantially large number of individual samples are
557 available. This result, which might basically stem from a more accurate estimation of allele
558 frequency when using PoolSeq data, echoes theoretical results in the comparative study by
559 Gautier et al. (2013).
560
561 **3.2 | Parameter estimation**
562 Table 2 shows point estimates with 90% credibility intervals of posterior distributions as
563 well as NMAE accuracy measures for the four parameters of interest r_a, t_1, N_4 and t_1/N_4.
564 NMAE values show that estimations were accurate both at the global and local scales for
565 the admixture rate r_a and the composite parameter t_1/N_4 (cf. the low NMAE values for
566 these parameters). In contrast to the composite parameter t_1/N_4, estimations of the original
567 parameters t_1 and N_4 were much less accurate (cf. higher NMAE values). This result is
568 also illustrated for the two pseudo-observed datasets by point estimates close to the true
569 values and narrow 90% CI for r_a and t_1/N_4. The low bias observed for r_a and t_1/N_4
570 estimates is satisfying considering that the training set only includes 10,000 datasets.
571 NMAE values computed from median point estimates were systematically smaller (albeit
572 sometimes only to a small extent) than those computed from mean point estimates,

indicating that the median is globally a better point estimate of the parameter than the mean. As expected when considering point estimates for the two pseudo-observed datasets, this trend did not translate for all parameters (e.g., the mean is slightly closer to the true value than the median for r_a).

We found that including PLS axes in the RF feature vector improved parameter estimation in a heterogeneous way. The accuracy gain of including PLS axes ranged from negligible (e.g. IndSeq global NMAE for t_1/N_4 based on median of 0.220 and 0.221 with and without PLS, respectively) to substantial (e.g. PoolSeq global NMAE for N_4 based on median of 0.380 and 0.421 with and without PLS, respectively). The accuracy levels were always lower at the global than local scales, sometimes to a large extent (e.g. PoolSeq NMAE for t_1/N_4 based on mean of 0.217 and 0.077 at the global and local scales, respectively).This illustrates how the accuracy estimation of at least some parameters can substantially differ depending on the location of an observed dataset in the prior data space. In the present case study, the pseudo-observed datasets are located in a favorable part of the prior space. Finally, like scenario choice analyses, we obtained considerably higher accuracy (i.e. lower NMAE values and this with or without PLS axes) for the PoolSeq dataset than the IndSeq dataset. This result is also illustrated by point estimates of the two pseudo-observed datasets closer to the true values and narrower ranges of 90% CI for PoolSeq than IndSeq for all parameters. This reinforces our previous conclusion that, for a similar sequencing effort, it is preferable to use a PoolSeq strategy than an IndSeq strategy when a large number of individual samples are available.

3.3 | Contribution to inferences of components of the feature vector

Learning more about how various summary statistics relate to scenarios or parameters would be useful for population genetics going forward. In the realm of standard ABC

methods, it is not clear which summary statistics are responsible for a signal. By contrast, many SML methods including RF allow direct measurement of the contribution of each component included in the feature vector. RF hence offer direct ways to assess which features of the input are driving inferences, information which can yield insights about the underlying processes (Breiman, 2001). ABC-RF therefore addresses some of the criticisms against the "black box" aspect of classical ABC methods. Figure 3 illustrates how RFs automatically rank the components of the feature vector (i.e., the various summary statistics plus the LDA or PLS axes when the latter metrics are added to the feature vector) according to their level of information when building trees of the forest. Figure 3 and Figure S2 show that informative statistics are different depending on the comparisons (individual scenarios or groups of scenarios) and the analyzed parameter under a given scenario. Four-sample and three-sample *f*-statistics, as well as the related three-sample coefficients of admixture (i.e. AML statistics), were among the most informative to discriminate scenarios (Figure 3A). In agreement with this, such statistics are by construction highly sensitive to the topology connecting populations including or not an admixture event (Patterson et al., 2012; Estoup et al., 2018). A typical feature of scenario choice RF analysis is that one or several LDA axes always correspond to the best informative statistics.

For parameter estimation, the most informative summary statistics were different depending on the parameter of interest (Figure 3B and Figure S2). Figure 3B shows that for the (well estimated) composite parameter t_1/N_4, the most informative statistics included three-sample *f*-statistics and AML statistics with the population 4 as target, the population-specific F_{ST}, ML1p (proportion of monomorphic loci) and heterozygosity - all for population 4 -, and pairwise-population statistics (F_{ST} and Nei's distance) that included population 4. For other parameter values, the set of informative statistics differed among

parameters, but always included a large number of four-sample and three-sample *f*-statistics, as well as three-sample AML statistics (Figure S3). In contrast to LDA axes (used for scenario choice), only a subset of PLS components were ranked among the 30 most informative statistics and they were never ranked at first position.

We added five noise variables (corresponding to values randomly drawn into uniform distributions bounded between 0 and 1) to the feature vector processed by RF in order to evaluate the threshold of variable importance metrics below which components of the vector were not informative anymore. We found that for both scenario choice and parameter estimation, a substantial proportion of summary statistics was not informative. For scenario choice, we found that only 28% to 38% of the summary statistics were informative. For parameter estimation, 20% to 65% of the summary statistics were informative. Non-informative statistics were different when considering scenarios by groups or separately, and depending on the parameter of interest (results not shown). It is worth stressing that such non-informative components of the feature vector are simply not or seldom chosen when constructing each individual trees of the forest, and hence do not alter RF inferences (Breiman, 2001; Marin et al., 2018; Raynal et al., 2019). In agreement with this, removing noise variables from the feature vector did not impact the levels of errors in scenario choice and of accuracy in parameter estimation in the present case study (results not shown).

4 | DISCUSSION

Population genetics is now poised for an explosion in the use of SML approaches (Schrider & Kern, 2018). In this context, any effort to create self-contained, efficient, and user-friendly software packages capable of performing the entire workflow associated to SML

methods would streamline such methods and make them more accessible to researchers, especially for non-specialist users. For this purpose, we developed the package DIYABC Random Forest v1.0 which integrates – within a user-friendly interface – a set of methods to simulate training sets for various types of molecular data under custom evolutionary scenarios, encode both the simulated and real data as large size feature vectors (summary statistics), train RF algorithms, apply them on observed data point(s), and assess their performance in term of prediction (using various metrics to evaluate error and accuracy). We illustrated the main potentialities and functionalities of DIYABC Random Forest v1.0 through the treatments of two example SNP pseudo-observed datasets. Our results indicate that SML methods such as RF show great promise in demographic estimation and scenario selection using genetic data and we argue that they may soon be the preferred choice over alternative methods based on classical ABC.

The first advantage of RF is that, given a pool of different metrics available (here various non-independent summary statistics and their linear combinations), this SML method extracts the maximum of information from the entire set of the proposed component of the feature vector. This avoids the arbitrary choice of a subset of components, which is often applied in ABC analyses, and also minimizes the curse of dimensionality whereby accuracy and stability of inferences decrease as the number of summary statistics grows (Beaumont et al., 2010; Blum et al., 2013). As a matter of fact, SML methods such as RF can handle many statistics, even if they are strongly correlated and/or unnecessary (i.e., virtually non-informative), with a limited impact on the performance of the method (Marin et al., 2018; Raynal et al., 2019). In practice, and in contrast to standard ABC methods, SML methods perform better when the input data have a large number of features, in what is commonly called the ‘blessing of dimensionality’ (e.g., Anderson et al., 2014; Breiman, 2001). In agreement with this, inputs that consist of

673 thousands of variables have been used with great success; e.g., Amit & Geman, 1997;
674 Chen et al., 2013; and unpublished results obtained using feature vectors of > 10,000
675 summary statistics to treat SNP datasets under complex evolutionary scenarios with
676 DIYABC Random Forest v1.0).

677 Regarding the composition of the feature vector, defining informative statistics to
678 be included in this vector remains an important issue of any SML method. We have
679 implemented a new set of summary statistics to better extract the genetic information
680 contained in the selectively neutral and independent SNP markers simulated in DIYABC
681 Random Forest v1.0. For both scenario choice and parameter estimation, our results show,
682 at least in the evolutionary contexts we explored, the high level of information content of
683 four-populations and three-populations *f*-statistics (Patterson 2012), - as well as the related
684 three-sample AML statistics (Cornuet et al., 2014). We found that inferences were more
685 accurate with this new set of SNP summary statistics than with the one previously
686 proposed in DYABC v2.1.0 (Cornuet et al., 2014). For instance, comparative treatments
687 based on the IndSeq example dataset, show that error levels were substantially lower and
688 accuracy higher with the new set of SNP summary statistics. More specifically, global
689 error rates were 13% and 5% lower when considering scenarios separately or by groups,
690 respectively, and global NMAE values – computed from the median – were 9% to 21%
691 lower depending on the estimated parameter (results not shown). The addition into the
692 feature vector of linear combinations of statistics (LDA and PLS axes for scenario choice
693 and parameter estimation, respectively) also globally improved our statistical inferences.
694 While the inferential gain was systematic and substantial for LDA axes, we found that
695 including PLS axes in the RF vector feature improved parameter estimation in a
696 heterogeneous way, with a negligible gain in some cases.

The second advantage of SML methods such as RF is that they naturally use all of the simulations to learn the mapping of data to scenarios and/or parameters. This contrasts to the rejection step of ABC which precludes an optimal use of the datasets that are not retained. This advantage remains although work has been done to retain more of the simulations in ABC, for instance by weighing their influence on parameter estimation according to their similarity to the observed data (e.g., Blum & Francois, 2010). Consequently, the computing effort is considerably reduced for RF, as the method requires a substantially smaller training set compared to ABC methods (e.g., a few thousands of simulated datasets versus hundreds of thousands of simulations per scenario for most ABC approaches; Blum & François, 2010; Fraimout et al., 2017; Pudlo et al., 2016; Raynal et al., 2019). Given the ever-increasing dimensionality of modern genetic data generated using NGS technologies, this is a particularly appealing property of SML methods. Moreover, it is worth noting that DIYABC Random Forest v1.0 relies on out-of-bag prediction to evaluate the error and accuracy of inferences, so that no additional potentially costly simulations of test datasets are necessary for this purpose.

RF is often considered as a “tuning-free” method in the sense that it does not require meticulous calibrations. This represents an important advantage of this method, especially for non-expert users. On the opposite, ABC methods require calibration to optimize their use, such calibration being time consuming when different levels of tolerance are tested and/or used. In practice, we nevertheless advise users to consider several checking points, thereafter formalized as questions, before finalizing inferential treatments using DIYABC Random Forest v1.0.

Are my scenarios and/or associated priors compatible with the observed dataset? This question is of prime interest and applies to ABC Random Forest as well as to any alternative ABC treatments. This issue is particularly crucial, given that complex scenarios

and high dimensional datasets (i.e., large and hence very informative datasets) are becoming the norm in population genomics. Basically, if none of the proposed scenario / prior combinations produces some simulated datasets in a reasonable vicinity of the observed dataset, this is a signal of incompatibility and it is not recommended to attempt any inferences. In such situations, we strongly advise reformulating the compared scenarios and/or the associated prior distributions in order to achieve some compatibility in the above sense. DIYABC Random Forest v1.0 proposes a visual way to address this issue through the simultaneous projection of datasets of the training set and of the observed dataset on the first LDA axes (e.g., Figure 2); see also other dedicated diagnostic tools in the notice of the program. In the LDA projection, the observed dataset has to be reasonably located within the clouds of simulated datasets.

Did I simulate enough datasets for my training set? A rule of thumb is, for scenario choice to simulate between 2,000 and 20,000 datasets per scenario among those compared (Pudlo et al., 2016; Estoup et al., 2018), and for parameter estimation to simulate between 10,000 and 100,000 datasets under a given scenario (Raynal et al., 2019; Chapuis et al., 2020). To evaluate whether or not this number is sufficient for RF analysis, we recommend to compute error/accuracy metrics such as those proposed by DIYABC Random Forest v1.0 from both the entire training set and a subset of the latter (for instance from a subset of 80,000 simulated datasets if the training set includes a total of 100,000 simulated datasets). If error (accuracy) metrics from the subset are similar, or only slightly higher (lower) than the value obtained from the entire database, one can consider that the training set contains enough simulated datasets. If a substantial difference is observed between both values, then we recommend increasing the number of simulated datasets in the training set.

Did my forest grow enough trees? According to our experience, a forest made of 500 to 2,000 trees often constitutes an interesting trade-off between computation efficiency

747 and statistical precision (Breiman, 2001; Chapuis et al., 2020; Pudlo et al., 2016, Raynal et
748 al., 2019). To evaluate whether or not this number is sufficient, we recommend plotting
749 error/accuracy metrics as a function of the number of trees in the forest. The shapes of the
750 curves provide a visual diagnostic of whether such key metrics stabilize when the number
751 of trees tends to a given value. DIYABC Random Forest v1.0 provides such a plot-figure
752 as output (e.g. Figure S1).

753 Various SML methods have been recently developed (e.g. Schrider & Kern, 2018).
754 In particular, neural networks are machine learning methods which are used increasingly in
755 population genetic, often under the term “deep learning” (Sheehan & Song, 2016), and
756 sometimes using an ABC framework (Mondal, Bertranpetit, & Lao, 2019). Deep learning,
757 with its incredibly flexible input and output structure, is expected to be an important area
758 of future research in many different fields including population genetics (e.g.
759 Angermueller et al., 2016; Schrider & Kern, 2018). For instance, rather than learning on
760 standard population genetic summary statistics calculated from SNP frequencies or
761 multiple sequence alignments, one can instead treat raw data such as the pixels of an image
762 of the sequence alignment itself as the input (Flagel, Brandvain, & Schrider, 2018). One of
763 the earliest application of deep learning, using a set of 345 traditional statistics describing
764 the SNP spectrum as input and considering a simple one-population scenario, has already
765 yielded the crucial ability to jointly infer demographic history and selection, a central goal
766 of population genetics analysis (Sheehan & Song, 2016). It is worth stressing, however,
767 that in contrast to RF, deep learning methods are not tuning-free and often require
768 meticulous calibrations, including the specification of the number of layers composing the
769 neural network, as well as thorough investigation of the regularization parameter of the
770 cost function. Moreover, deep learning methods require datasets of larger size and
771 substantially larger computing resources than RF. We hence believe that RF remains one

of the most competitive SML methods when no tuning of parameters is desired. The RF method remains particularly attractive for non-expert machine-learning users, especially when it is embedded in an integrative user-friendly interfaced program such as DIYABC Random Forest 1.0.

In conclusion, although SML approaches are revolutionizing many fields, their use in population genetics inference is still in its infancy (Schrider & Kern 2018). However, the recent successes of SML approaches in the latter scientific field demonstrate that they have the potential to revolutionize the practice of population genetic data analysis. In particular, SML methods such as RF may soon be the preferred choice over ABC method in scenario selection and demographic estimation, especially when analyzing multiple complex scenarios and large-size datasets. In this context, DIYABC Random Forest v1.0 provides an integrative operational solution streamlining the entire workflow to applying RF methods to various types of population genetic data. We believe that, because of the general properties of the implemented RF methods and the large set of summary statistics available for SNP data, DIYABC Random Forest v1.0 represents a useful resource to make efficient inferences about population genetic history from high dimensional genetic dataset, as typically obtained from NGS technologies. More generally, the RF methodologies we propose should appeal to all scientific fields in which big datasets are generated under complex scenarios using simulation-based methods and summarized under a large-size feature vector.

ACKNOWLEDGEMENTS

This work was supported by funds from the French Agence National pour la Recherche (ANR projects SWING and GANDHI), the INRAE scientific division SPE (AAP-SPE 2016), and the LabEx NUMEV (NUMEV, ANR10-LABX-20). We thank Pierre Pudlo for

797 useful discussions, and Jean-Marie Cornuet and Alex Dehne Garcia for computer code
798 expertise at the onset of the ABC Random Forest project.
799
800

REFERENCES

Amit Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Compution*, 9, 1545–1588

Anderson, J., Belkin, N., Goyal, L., Rademacher & Voss, J. (2014). The more, the merrier: the blessing of dimensionality for learning large Gaussian mixtures. *Proceedings of The 27th Conference on Learning Theory, PMLR 35,* 1135-1164

Angermueller, C., Parnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology, 12(7),* 878. https://doi.10.15252/msb.20156651

Beaumont, M.A., Zhang, W., & Balding, D.J. (2002). Approximate Bayesian Computation in population genetics. *Genetics, 162(4),* 2025–2035. PMID: 12524368

Beaumont, M. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics, 41(1)*, 379–406. https://doi.org/10.1146/annurev-ecolsys-102209-144621

Bertorelle, G., Benazzo, A., & Mona, S. (2010). ABC as a flexible framework to estimate demography over space and time: some cons, many pros. *Molecular Ecology, 19(13*), 2609–2625. https://doi.10.1111/j.1365-294X.2010.04690.x

Blum, M.G.B., & François, O. (2010).Non-linear regression models for Approximate Bayesian Computation. *Statistics and computing*. *20(1*):63–73. https://doi.10.1007/s11222-009-9116-0

Blum, M.G.B., Nunes, M.A., Prangle, D., & Sisson, SA. (2013). A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science, 28(2*), 189–208. https://doi.10.1214/12-STS406

Breiman, L. (2001). Random forests. *Machine Learning, 45(1),* 5–32. https://doi.org/10.1023/A:1010933404324

Chang, W., Cheng, J., Allaire, JJ, Xie, Y., & McPherson J. (2019). Shiny: Web Application Framework for R. R package version 1.4.0. https://CRAN.R-project.org/package=shiny

Chapuis, M-P R., Raynal, L., Plantamp, C., Meynard, CN., Blondin, L., Marin, J-M., & Estoup, A. (2020). A young age of subspecific divergence in the desert locust *Schistocerca gregaria*, inferred by ABC Random Forest, bioRxiv, 671867, ver. 4 peer-reviewed by *Peer Community in Evolutionary Biology*. https://www.biorxiv.org/content/10.1101/671867v4

Chen, D., Cao, X., Wen, F., & Sun J. (2013). Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 3025–3032, IEEE

Cornuet, J.-M., Pudlo, P., Veyssier, J., Dehne-Garcia, A., Gautier, M., Leblois, R.,…., & Estoup, A. (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),* 1187–1189. https://doi.10.1093/bioinformatics/btt763

Csilléry, K., Blum, M. G., Gaggiotti, O. E., & François, O. (2010). Approximate Bayesian Computation (ABC) in practice. *Trends in Ecology & Evolution, 25(7),* 410–418. https://doi.10.1016/j.tree.2010.04.001

Estoup, A., Lombaert, E., Marin, J.-M., Robert, C., Guillemaud, T., Pudlo, P., & Cornuet, J.-M. (2012). Estimation of demo-genetic model probabilities with Approximate Bayesian Computation using linear discriminant analysis on summary statistics.

Molecular Ecology Ressources, 12(5), 846–855. https://doi.10.1111/j.1755-0998.2012.03153.x

Estoup, A., Raynal, L., Verdu, P., & Marin, J-M. (2018) Model choice using Approximate Bayesian Computation and Random Forests: analyzes based on model grouping to make inferences about the genetic history of Pygmy human populations. *Journal de la Société Française de Statistiques, 159(3),* 167-190

Flagel, L., Brandvain, Y., & Schrider, D. R. (2018). The unreasonable effectiveness of convolutional neural networks in population genetic inference. *Molecular Biology and Evolution, 36(2),* 220-238. https://doi: 10.1093/molbev/msy224

Fraimout, A., Debat, V., Fellous, S., Hufbauer, R. A., Foucaud, J., Pudlo, P.,…Estoup A. (2017). Deciphering the routes of invasion of *Drosophila suzukii* by means of ABC Random Forest. *Molecular biology and evolution, 34(4),* 980–996. https://doi.10.1093/molbev/msx050

Gautier, M., Foucaud, J., Gharbi, K., Cezard, T., Galan, M., Loiseau, A., …, Estoup, A. (2013) .Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology, 22(14),* 3766-3779. http://doi.10.1111/mec.12360

Hivert, V., Leblois, R., Petit, E. J., Gautier, M., & Vitalis, R (2018). Measuring genetic differentiation from pool-seq data. *Genetics, 210(1):* 31-330. https://doi.10.1534/genetics.118.300900

Hudson, R. R. (1993). The how and why of generating gene genealogies. In Takahata, N. & Clark, A. G., editors, *Mechanisms of Molecular Evolution*, pages 23–36. Sinauer Associates, Sunderland, MA.

Hudson, R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics, 18(2),* 337-338. https://doi.org/10.1093/bioinformatics/18.2.337

Leblois, R., Gautier M., Rohfritsch A., Foucaud J., Burban C., Galan M.,…,Kerdelhué C. (2018). Deciphering the demographic history of allochronic differentiation in the pine processionary moth *Thaumetopoea pityocampa*. *Molecular Ecology, 27(1),* 264-278. doi.10.1111/mec.14411

Libbrecht, M.W. & Noble, W.S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics. 16(6):321-32.* https://doi.10.1038/nrg3920

Marin, J.-M., Pudlo, P., Estoup, A., & Robert, C. P. (2018). Likelihood-free model choice. in handbook of approximate Bayesian computation. In Sisson, S., Fan, Y., & Beaumont, M., editors, *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC. https://doi.org/10.1002/bimj.201900141

Meinshausen, N. (2006) Quantile regression forests. *Journal of Machine Learning Research*, 7, 983–999. https://dl.acm.org/doi/10.5555/1248547.1248582

Mondal, M., Bertranpetit, J., & Lao, O. (2019). Approximate Bayesian Computation with deep learning supports a third archaic introgression in Asia and Oceania. *Nature Communications, 10,* 246. https://doi.org/10.1038/s41467-018-08089-7

Nei, M. (1972). Genetic distance between populations. *American Naturalist, 106,* 283-292. https://doi.org/10.1086/282771

Patterson, N., Moorjani, P., Luo, Y., Mallick, S., Rohland, N., Zhan, Y., & Reich, D. (2012). Ancient admixture in human history. *Genetics, 192 (3),* 1065–1093. https://doi.org/10.1534/genetics.112.145037

Peter B.M. (2016). Admixture, population structure, and F-statistics. *Genetics, 202(4),* 1485-1501. https://doi:10.1534/genetics.115.183913

Pudlo, P., Marin, J.-M., Estoup, A., Cornuet, J.-M., Gautier, M., & Robert, C. P. (2016). Reliable ABC model choice via random forests. *Bioinformatics, 32(6),* 859–866. https://doi.10.1093/bioinformatics/btv684

Pybus, M., Dall'Olio, GM., Luisi, P.,Uzkudun, M., Laayouni., H., Bertranpetit, J., & Engelken., J. (2015). Hierarchical boosting: a machine-learning framework to detect and classify hard selective sweeps in human populations. *Bioinformatics, 31(24),* 3946–3952. https://doi.org/10.1093/bioinformatics/btv493

Raynal L., Marin J-M., Pudlo P., Ribatet M., Robert C.P., & Estoup A (2019). ABC random forests for Bayesian parameter inference. *Bioinformatics, 35(10),* 1720–1728. https ://doi.10.1093/bioinformatics/bty867

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34(1),* 1–47. https://doi.org/10.1145/505282.505283

Schlötterer, C., Tobler, R., Kofler, R., & Nolte, V. (2014). Sequencing pools of individuals - mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics, 15(11),* 749-763. doi.10.1038/nrg3803

Schrider, D.R. & Kern, A.D. (2016). S/HIC: robust Identification of soft and hard sweeps using machine learning. *PLoS Genetics*, *12(3),* e1005928. https://doi.org/10.1371/journal.pgen.1005928

Schrider D.R., Ayroles J., Matute D.R., & Kern A.D. (2018). Supervised machine learning reveals introgressed loci in the genomes of *Drosophila simulans* and *D. sechellia*. *PLoS Genetics, 14(4)*: e1007341. https://doi.org/10.1371/journal.pgen.1007341.

Schrider, D.R., & Kern, A.D. (2018). Supervised machine learning for population genetics: a new paradigm. *Trends in* Genetics, *34(4),* 301-312. https://doi.org/10.1016/j.tig.2017.12.005

Sheehan S., & Song Y.S. (2016). Deep Learning for Population Genetic Inference. *PLoS Computional Biology, 12(3)*: e1004845. https ://doi:10.1371/journal. pcbi.1004845

Weir, BS., & Goudet, J. (2017). A unified characterization of population structure and relatedness. *Genetics*, *206(4),* 2085-2103. https ://doi.10.1534/genetics.116.198424

Wickham, H. (2016) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. ISBN 978-0-387-98141-3

Wright, M. N. & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software, 77(1),* 1-17. https ://doi.10.18637/jss.v077.i01

DATA ACCESSIBILITY

For the two PoolSeq and IndSeq example datasets used in this paper, we provide the corresponding pseudo-observed datasets (read numbers or genotype data and summary statistics: i.e. <file_name.snp> and statobsRF.txt), the headerRF files (headerRF.txt) and the training set files (reftableRF.bin) at https://github.com/diyabc/diyabc/tree/master/diyabc-tests/MER

AUTHOR CONTRIBUTIONS

Conceptualization, F-D.C., L.R., J-M.M. and A.E.; Core program coding: F-D.C; Interface coding: G.D; New SNP summary statistics: M.G., R.V and A.E.; Program debugging and testing: F-D.C, E.L. and A.E.; Example datasets analysis, A.E,; Writing – Original Draft, A.E.; Writing – Review & Editing, F-D.C., L.R., G.D., M.G., R.V., E.L., J-M.M. and A.E.; Funding acquisition, J-M.M. and A.E.

TABLE 1. Results for scenario choice.

The six compared scenarios and the two groups of scenarios are detailed in Figure 1. The parameter values for the two example pseudo-observed datasets are: N_1=7,000, N_2=2,000, N_3=4,000, N_4=3000, t_1=200, r_a=0.3, t_2=300 and t_3=500. RF analyses used a training set including feature vector values from 12,000 simulated datasets (2,000 per scenario) and the number of trees was 1,000. Global (prior) and local (posterior) error rates were estimated using out-of-bag estimators from a sample of 10,000 data randomly chosen in the training set. Standard deviations over the ten replicate analyses are given between brackets for each metrics, in addition to the means. In the "With LDA" treatments, five LDA axes were added to the set of 130 summary statistics composing the feature vector.

Type of dataset	Type of treatment		Global error rate	Local error rate	Vote scen. 1	Vote scen. 2	Vote scen. 3	Vote scen. 4	Vote scen. 5	Vote scen. 6	Posterior probability
PoolSeq	Groups of scenarios: with vs. without admixture	With LDA	0.172 (0.001)	0.085 (0.009)		925.1 (10.754)			74.9 (10.754)		0.915 [group 1] (0.009)
		Without LDA	0.192 (0.001)	0.162 (0.015)		891.9 (13.585)			108.1 (13.585)		0.838 [group 1] (0.015)
	All scenarios considered separately	With LDA	0.196 (0.0008)	0.135 (0.011)	4.1 (1.297)	65.3 (7.364)	897.0 (8.056)	8.5 (2.121)	15.7 (4.808)	9.4 (2.011)	0.865 [scen. 3] (0.011)
		Without LDA	0.220 (0.0009)	0.202 (0.013)	9.3 (3.020)	98.5 (17.264)	829.4 (20.304)	8.3 (2.669)	32.9 (3.381)	21.6 (4.671)	0.798 [scen. 3] (0.013)
IndSeq	Groups of scenarios: with vs. without admixture	With LDA	0.212 (0.001)	0.177 (0.016)		840.6 (12.816)			159.4 (12.816)		0.823 [group 1] (0.016)
		Without LDA	0.220 (0.002)	0.270 (0.016)		805.5 (18.940)			194.5 (18.940)		0.730 [group 1] (0.016)
	All scenarios considered separately	With LDA	0.248 (0.001)	0.268 (0.018)	6.9 (3.107)	105.9 (10.692)	817.0 (13.021)	12.7 (3.057)	41.2 (5.159)	16.3 (3.653)	0.732 [scen. 3] (0 .018)
		Without LDA	0.262 (0.001)	0.343 (0.0197)	9.0 (2.981)	123.5 (7.322)	769.6 (12.366)	15.8 (4.049)	60.7 (8.982)	21.4 (4.274)	0.657 [scen. 3] (0.020)

TABLE 2. Results for estimation of parameters of interest under scenario 3.

True values of parameters of interest for the two example pseudo-observed datasets are r_a = 0.3, t_1 = 200, N_4 = 3,000 and t_1/N_4 = 0.067. RF analyses used a training set including feature vector values from 10,000 simulated datasets and the number of trees was 1,000. Global (prior) and local (posterior) NMAE values were estimated using out-of-bag estimators from a sample of 10,000 data randomly chosen in the training set. Standard deviations over the ten replicate analyses are given between brackets for each metrics, in addition to the means. In the "With PLS" treatments, the number of PLS axes which were added to the set of 130 summary statistics of the feature vector for the PoolSeq (IndSeq) datasets was equal to 12 (12), 18 (21), 23 (24), and 4 (4) for r_a, t_1, N_4 and t_1/N_4, respectively. CI: credibility interval.

Type of dataset	Type of treatment	Parameter	Posterior point estimates of			Global (prior) NMAE computed from		Local (posterior) NMAE computed from	
			Mean	Median	90% CI	Mean	Median	Mean	Median
PoolSeq	With PLS	r_a	0.346 (0.0018)	0.352 (0.0030)	0.248 - 0.422 (0.0041) (0.0040)	0.133 (0.0002)	0.123 (0.0002)	0.0890 (0.0028)	0.0887 (0.0024)
		t_1	291.4 (3.366)	300.5 (2.273)	147.6 - 441.0 (3.777) (3.887)	0.312 (0.0003)	0.290 (0.0003)	0.2022 (0.0047)	0.1999 (0.0045)
		N_4	4040 (37.16)	3658 (58.55)	1861 - 7399 (90.42) (161.6)	0.416 (0.0005)	0.380 (0.0006)	0.3169 (0.0094)	0.2848 (0.0093)
		t_1/N_4	0.067 (0.0004)	0.068 (0.0005)	0.049 - 0.084 (0.0010) (0.0006)	0.217 (0.0008)	0.178 (0.0002)	0.0786 (0.0020)	0.0773 (0.0016)
	Without PLS	r_a	0.364 (0.0028)	0.368 (0.0032)	0.245 - 0.483 (0.0072) (0.0055)	0.143 (0.00026)	0.130 (0.00028)	0.0996 (0.0032)	0.0979 (0.0025)
		t_1	288.2 (3.752)	301.7 (2.540)	134.4 - 443.0 (8.044) (4.546)	0.322 0.00042)	0.301 0.00046)	0.2354 (0.0095)	0.2311 (0.0091)
		N_4	5517 39.60	5539 94.16	2319 - 8662 104.3 133.8	0.456 0.0006	0.421 0.0006	0.3243 (0.0116)	0.2972 (0.0091)
		t_1/N_4	0.068 (0.0004)	0.068 (0.0006)	0.050 - 0.085 (0.0008) (0.0006)	0.218 (0.0009)	0.179 (0.0003)	0.0789 (0.0021)	0.0781 (0.0017)
IndSeq	With PLS	r_a	0.402 (0.0041)	0.391 (0.0040)	0.275 - 0.611 (0.0041) (0.0096	0.172 (0.0003)	0.154 (0.0003)	0.1605 (0.0021)	0.1504 (0.0020)
		t_1	400.5 (3.133)	395.6 (2.875)	231.5 - 574.1 (4.478) (11.083)	0.398 (0.0006)	0.357 (0.0006)	0.1793 (0.0056)	0.1791 (0.0051)
		N_4	6608 (53.15)	6796 (55.61)	2861 - 9513 (111.6) (148.7)	0.476 (0.0006)	0.442 (0.0007)	0.2494 (0.0117)	0.2485 (0.0105)

		t_1/N_4	0.061 (0.0004)	0.061 (0.0004)	0.044 - 0.077 (0.0006) (0.0009)	0.262 (0.0009)	0.220 (0.0006)	0.0910 (0.0025)	0.0901 (0.0025)
	Without PLS	r_a	0.417 (0.0052	0.410 (0.0041	0.284 - 0.612 (0.0050) (0.0143	0.173 (0.0003	0.155 (0.0003)	0.1624 (0.0055	0.1526 (0.0048
		t_1	399.0 (3.184)	395.2 (3.370)	227.9 - 591.9 (4.653) (4.758)	0.407 (0.0005)	0.366 (0.0005)	0.1910 (0.0042)	0.1899 (0.0041)
		N_4	5837 (50.70)	5978 (93.02)	2524 - 9210 (134.6) (166.8)	0.499 (0.0006)	0.467 (0.0006)	0.2952 (0.0053)	0.2917 (0.0051)
		t_1/N_4	0.061 (0.0003)	0.061 (0.0003)	0.045 - 0.078 (0.0005) (0.0008)	0.263 (0.0008)	0.221 (0.0005)	0.0921 (0.0025)	0.0915 (0.0024)

FIGURE 1. Evolutionary scenarios compared.
The target population (pop 4) has three possible single (i.e., non-admixed) population sources (pop 1, pop 2 or pop 3) composing a group of three scenarios without admixture (group 2 in the figure) and three possible admixed pairwise population sources (i.e., admixture between pop1& pop2, pop 1& pop3 and pop 2 & pop3) composing a group of three scenarios without admixture (group 1 in the figure). Demographic and historical parameters include four effective population sizes N_1, N_2, N_3 and N_4 (for populations 1, 2, 3, and 4, respectively) and three divergence or admixture time events (t_1, t_2 and t_3), For the scenarios with admixture, the parameter r_a corresponds to the proportion of genes of a given source population entering into the admixed population 4. See text for details about prior distribution of parameters.

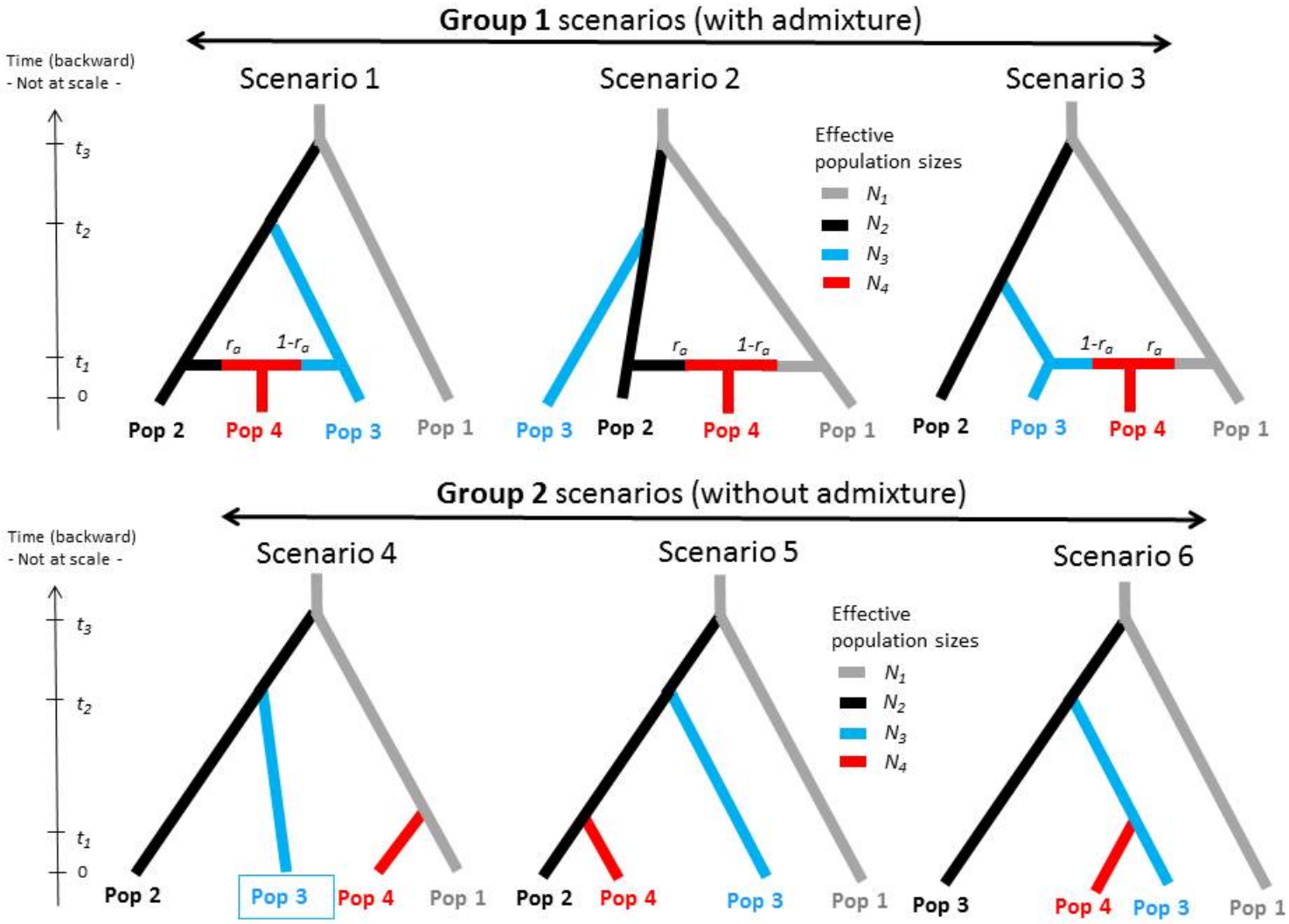

Group 1 scenarios (with admixture)
Time (backward)
- Not at scale -
Scenario 1
Scenario 2
Scenario 3
t_3
t_2
t_1
0
r_a
$1-r_a$
Effective population sizes
N_1
N_2
N_3
N_4
Pop 2
Pop 4
Pop 3
Pop 1
Group 2 scenarios (without admixture)
Scenario 4
Scenario 5
Scenario 6

FIGURE 2. Projection of the PoolSeq datasets from the training set on a single LDA axis when analyzing the two groups of scenarios (A) or on the first two LDA axes when analyzing the six scenarios separately (B).
The location of the pseudo-observed dataset in the LDA projection is indicated by a vertical line and a star symbol in panels A and B, respectively.

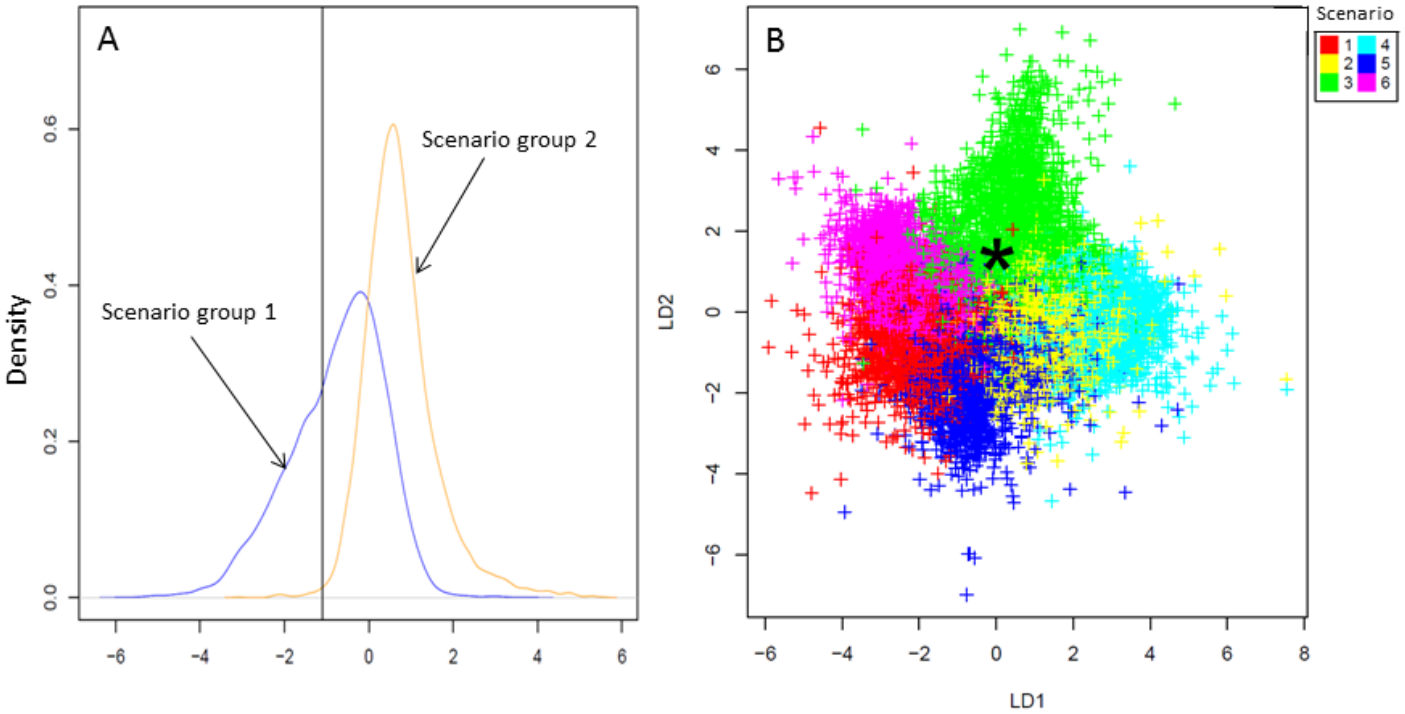
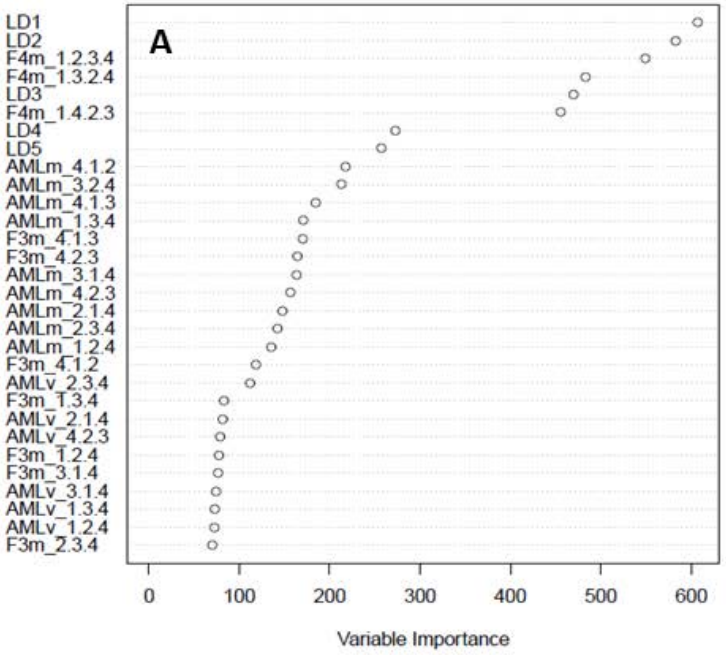
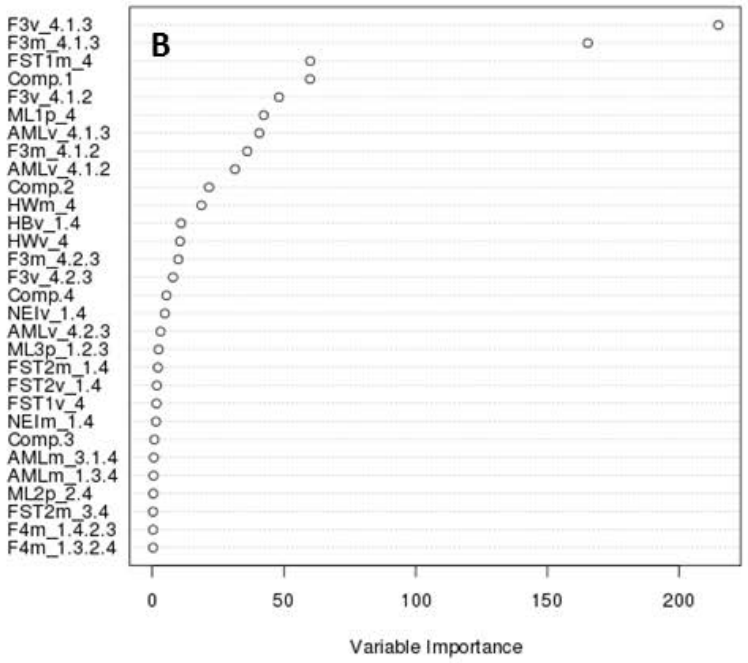

A
Density
Scenario group 1
Scenario group 2
B
LD1
LD2
Scenario
1
2
3
4
5
6

FIGURE 3. Contributions for the PoolSeq data analyses of the 30 most informative statistics to the Random Forest when choosing among scenarios considered separately (A) and when estimating the parameter t_1/N_4 under scenario 3 (B).
The variable importance of each statistics is computed as the mean decrease of impurity across the trees, where the impurity measure is the Gini index, and the residual sum of squares for scenario choice and parameter inference, respectively. For each variable, the sum of the impurity decrease across every tree of the forest is accumulated every time that variable is chosen to split a node. The sum is divided by the number of trees in the forest to give an average. The scale is irrelevant: only the relative values matter. The variable importance was computed for each of the 130 summary statistics provided by DIYABC Random Forest, plus the LDA axes for scenario choice (denoted LD) or the PLS axes for parameter estimation (denoted Comp.) that were added to the feature vector. The higher the variable importance the more informative is the statistic. Population index(s) are indicated at the end of each statistics and correspond to those in Figure 1. More details about summary statistics can be found in Table S1. See Figure S3 for an illustration of the contributions of the most informative statistics when choosing among the two groups of scenarios and when estimating the parameters r_a, t_1 and N_4.


A
LD1
LD2
F4m_1.2.3.4
F4m_1.3.2.4
LD3
F4m_1.4.2.3
LD4
LD5
AMLm_4.1.2
AMLm_3.2.4
AMLm_4.1.3
AMLm_1.3.4
F3m_4.1.3
F3m_4.2.3
AMLm_3.1.4
AMLm_4.2.3
AMLm_2.1.4
AMLm_2.3.4
AMLm_1.2.4
F3m_4.1.2
AMLv_2.3.4
F3m_1.3.4
AMLv_2.1.4
AMLv_4.2.3
F3m_1.2.4
F3m_3.1.4
AMLv_3.1.4
AMLv_1.3.4
AMLv_1.2.4
F3m_2.3.4
0
100
200
300
400
500
600
Variable Importance


B
F3v_4.1.3
F3m_4.1.3
FST1m_4
Comp.1
F3v_4.1.2
ML1p_4
AMLv_4.1.3
F3m_4.1.2
AMLv_4.1.2
Comp.2
HWm_4
HBv_1.4
HWv_4
F3m_4.2.3
F3v_4.2.3
Comp.4
NEIv_1.4
AMLv_4.2.3
ML3p_1.2.3
FST2m_1.4
FST2v_1.4
FST1v_4
NEIm_1.4
Comp.3
AMLm_3.1.4
AMLm_1.3.4
ML2p_2.4
FST2m_3.4
F4m_1.4.2.3
F4m_1.3.2.4
0
50
100
150
200
Variable Importance