GADMA2: more efficient and flexible demographic inference from genetic data

Ekaterina Noskova[1,⋆], Nikita Abramov[2], Stanislav Iliutkin[1], Anton Sidorin[3], Pavel Dobrynin[1,†], and Vladimir Ulyantsev[1,†]

[1]Computer Technologies Laboratory, ITMO University
[2]HSE University
[3]Laboratory of Biochemical Genetics, Department of Genetics and Biotechnology, St. Petersburg State University
[⋆]Denotes corresponding author
[†]Denotes shared senior authorship, listed alphabetically

Abstract

Inference of complex demographic histories typically requires parameterized models specified manually by the researcher. With an increased variety of methods and tools, each with its own interface, model specification becomes tedious and error-prone. Moreover, optimization algorithms used to find optimal parameters sometimes turn out to be inefficient. The open source software GADMA addresses these problems, providing automatic demographic inference. It proposes a common interface for several simulation engines and provides global optimization of parameters based on a genetic algorithm. Here, we introduce new features of GADMA2, the second version of the GADMA software. It has renovated core code base, new simulation engines, an updated optimization algorithm, and flexible specification of demographic history parameters. We provide a full overview of GADMA2 enhancements and demonstrate example of their usage.

1 Introduction

Genetic variation of closely-related populations and species is formed by evolutionary forces. Principal historical events like divergence, changes in population size, migration and selection could be reconstructed from the genetic data using different algorithmic and statistical approaches. Inference of complex demographic historyis widely applied in conservation biology studies to identify major events in population history. Moreover, it supplements archaeological information about the historical events that have left no paleontological records. Moreover, demographic histories can form the basis for subsequent population studies and medical genetic research.

In recent years many methods for demographic inference have appeared to investigate the demographic histories of species populations from genomes of individuals (Gutenkunst et al., 2009, Jouganous et al., 2017, Steinrücken et al., 2019, Kamm et al., 2020, Excofffier et al., 2021, DeWitt et al., 2021). Most of these provide means to simulate data statistics under a proposed, user-defined demographic history and compare them with real data by some measure of similarity. Thus, demographic inference is an estimation of model parameters with different likelihood-based optimization algorithms. One of the most widely-used data statistics is the allele frequency spectrum (e.g. Gutenkunst et al. (2009), Jouganous et al. (2017), Kamm et al. (2020)). However, newer methods based on two-locus (Ragsdale and Gutenkunst, 2017) and linkage disequilibrium statistics

(Ragsdale and Gravel, 2019, 2020) have also become available. While a number of different optimization techniques are used to identify maximal likelihood demographic parameters, they often turn out to be ineffective in practical applications (Noskova et al., 2020).

In 2020 we presented a new software GADMA (Noskova et al., 2020) for unsupervised demographic inference from allele frequency spectrum (AFS) data. GADMA separates the simulation and optimization units. It provides a common interface for various simulation engines and new global search optimization based on a genetic algorithm. It was shown that the proposed method has better performance than previously existing optimization algorithms both on simulated and real datasets (Noskova et al., 2020). Since its initial publication, GADMA has been applied in several studies on a variety of species: Xiong et al. (2021), Valdez and D'Elía (2021), Pazhenkova and Lukhtanov (2021), Cassin-Sackett et al. (2021), Buggiotti et al. (2021).

The initial version of GADMA featured only two simulation engines: ∂a∂i (Gutenkunst et al., 2009) and *moments* (Jouganous et al., 2017). Both of these engines work with allele frequency spectrum statistics and provide similar results. Among the variety of other tools that are available, we can highlight some other popular methods based either on AFS (*momi2*, *fastsimcoal2*), LD statistics (*momentsLD*) or haplotype data (diCal2) as potential valuable additions to the supported engines in GADMA. Moreover, both ∂a∂i and *moments* have been upgraded since these programs were first published and since GADMA's initial release. For example, ∂a∂i introduced inference of inbreeding coefficients (Blischak et al., 2020) and has started to support demographic histories involving four and five populations and enabled GPU support (Gutenkunst, 2021). In light of these advancements, we have sought to extend GADMA in several directions to support new engines and further enhance its optimization algorithms. In this paper we describe new capabilities implemented in GADMA2. The improved version has updated core codebase and implements a more efficient and flexible unsupervised demographic inference method.

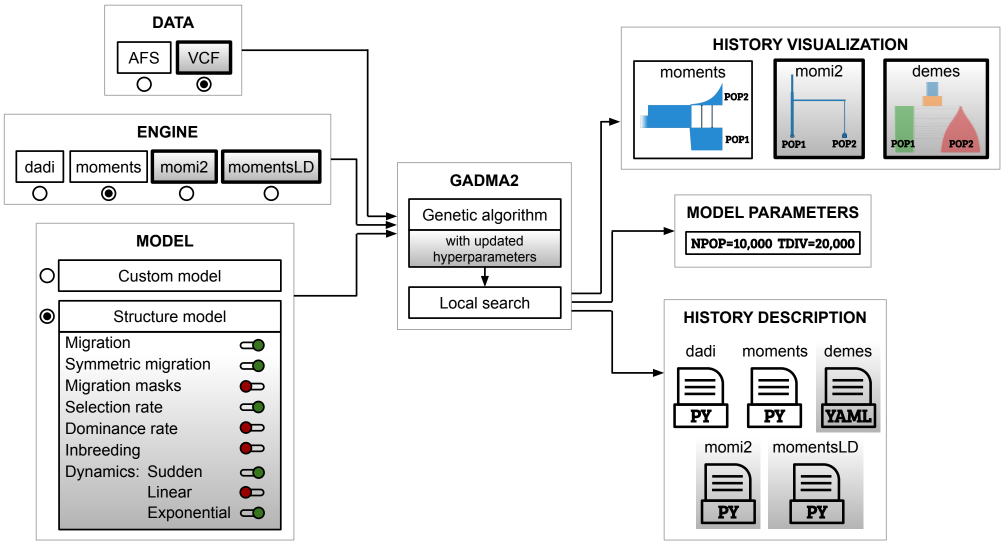


Figure 1: Scheme of GADMA2. New features and enhancements are marked with gradient grey colour. GADMA2 takes input genetic data presented in either AFS or VCF formats, engine name and model specifications and provides inferred model parameters, visualization and descriptions of final demographic history.

2 Results

GADMA2 extends the initial version in several ways (Figure 1). First, the genetic algorithm used in GADMA was improved with hyperparameter optimization. We have received new values of genetic algorithm hyperparameters that fall into more efficient and stable convergence. Second, GADMA2 provides more flexible control on model specification for automatic model construction. For example, it is possible to include inferences about selection and inbreeding coefficients. Third, two new simulation engines were integrated: *momi2* and *momentsLD*. Thus, GADMA2 now supports four engines overall. Lastly, several useful enhancements were integrated including the ability to use data in VCF format and new engines for model representations and visualization (*momi2* and *demes*).

2.1 Updated genetic algorithm

The method for demographic parameters estimation in GADMA is based on a genetic algorithm (Noskova et al., 2020). A *hyperparameter* is usually defined as a parameter of an algorithm. The performance of any algorithm depends on its hyperparameters and optimization of their values can significantly improve the overall efficiency. As an example of a hyperparameter, we can consider the number of demographic models in one iteration of the genetic algorithm. Several techniques can be used for optimization of hyperparameters and Bayesian optimization is a primary method among them (Snoek et al., 2012). The most popular and efficient method based on Bayesian optimization that performs hyperparameter optimization on the proposed set of problem instances is implemented in SMAC software (Hutter et al., 2011, Lindauer et al., 2021). It has been applied in a number of studies including optimization of neural networks (Lago et al., 2018, Hewamalage et al., 2021, Wu et al., 2022).

We used SMAC to tune hyperparameters of the genetic algorithm in GADMA2. The descrip-

Table 1: Values of the genetic algorithm hyperparameters after each round of their optimization with SMAC. Round 1 hyperparameter values are equal to the default values as SMAC failed to find better configuration. For each round of 2-6 rounds two discrete hyperparameters (`gen_size` and `n_init_const`) were fixed in order to gain SMAC efficiency and achieve a grid search.

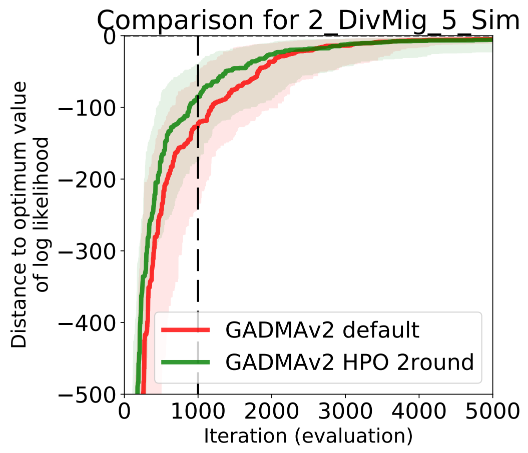
Hyperparameter ID	Round number					
	1 (default)	2	3	4	5	6
`gen_size`	10	10*	10*	10*	50*	50*
`n_init_const`	10	10*	5*	20*	10*	20*
`p_elitism`	0.20	0.30	0.30	0.40	0.40	0.40
`p_mutation`	0.30	0.20	0.20	0.10	0.08	0.10
`p_crossover`	0.30	0.30	0.30	0.30	0.42	0.46
`p_random`	0.20	0.20	0.20	0.20	0.10	0.04
`mutation_strength`	0.200	0.776	0.370	0.534	0.833	0.528
`const_mutation_strength`	1.010	1.302	1.290	1.648	1.199	1.492
`mutation_rate`	0.200	0.273	0.886	0.882	0.595	0.345
`const_mutation_rate`	1.020	1.475	1.942	1.417	1.645	1.472

*These values were fixed during the hyperparameter optimization with SMAC.

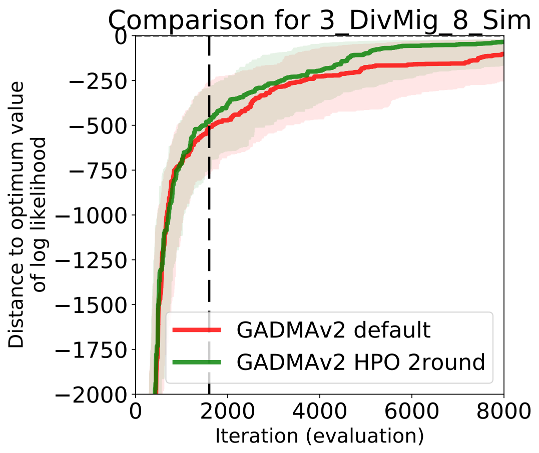
tions and domains of all hyperparameters can be found in Table 4 and Table 5 (see the Materials and Methods). Ten hyperparameters (Table 1) of the genetic algorithm were optimised during the first round of SMAC. SMAC performed 13,900 runs of the genetic algorithm and tested 2,222 different hyperparameter configurations. This process took two weeks of continuous computations. However, it failed to find a better solution than the default one. We assumed that such behaviour may be caused by the presence of two discrete hyperparameters (`gen_size` and `n_init_const`) in the configuration. These hyperparameters of the genetic algorithm were fixed to several specific values for the next rounds of SMAC optimization in order to perform a grid search.

We performed six rounds of SMAC optimization for different configurations of hyperparameters. Final configurations obtained from SMAC are presented in Table 1. The optimization with SMAC was performed for the *moments* simulation engine. The final solutions were also validated on all datasets using the *moments*, ∂a∂i and *momi2* engines. The mean values of log-likelihood were independently evaluated from 128 runs on training and test datasets in a fixed number of log-likelihood evaluations that were used in SMAC. They can be found in Table A1 for the *moments* engine, Table A2 for the ∂a∂i engine and Table A3 for the *momi2* engine. The costs and results for ∂a∂i are very similar to those shown in Table A1 with runs executed using the *moments* engine. These results support the idea that the ∂a∂i and *moments* engines have very similar performance.

Configuration from round 2 showed better and faster convergence for all three tested engines when compared to other configurations on most datasets. Thus, hyperparameters from round 2 were chosen as new updated hyperparameters for the genetic algorithm in GADMA2 (Figure 2). However, we note that according to convergence plots on greater number of iterations for *moments*



(a)



(b)

Figure 2: Example convergence plots for default genetic algorithm configuration from the initial version of GADMA (red colour) and configuration obtained during rounds 2 of hyperparameter optimization with SMAC (green colour) on two datasets: (a) train dataset `2_DivMig_5_Sim`, (b) test dataset `3_DivMig_8_Sim`. The abscissa presents the log-likelihood evaluation number, the ordinate refers to the distance to the optimal value of log-likelihood. Solid lines correspond to median convergence over 128 runs and shadowed areas are ranges between first (0.25) and third (0.75) quartiles. The vertical dashed black line refers to the number of evaluations used to stop a genetic algorithm in SMAC.

engine the configuration from round 6 demonstrated better median convergence and more narrow confidence intervals on some datasets (Figure A1 and Figure A2). Configuration from round 4 also demonstrated good convergence on several datasets both for ∂a∂i and *momi2* engine. More information and details are available in the Materials and Methods section.

2.2 Flexible demographic model

Automatic demographic model construction is a crucial feature of GADMA. The requirement of model choice that is necessary for classical tools is replaced by specification of the model structure that determines how detailed the model will be. The structure of the demographic model defines the number of model epochs before and between population splitting events. The demographic history is constructed automatically and all possible parameters are estimated. However, the researcher can control model parameter settings if desired or necessary. As a result, GADMA2 has a more flexible regulation on model construction.

Migration rates

One of the existing controls over model parameters in the initial version of GADMA was the opportunity to disable all migration events and to infer demographic history without any gene flow. GADMA2 now includes a new control handle to make migrations symmetric. Additionally, it allows for specific migrations to be disabled by setting up migration masks.

Selection and dominance rates

Both of the initially supported simulation engines included in GADMA, ∂a∂i and *moments*, are able to infer selection and dominance rates. The first version of GADMA lacked the function to make these inferences, but we have added these in the new version. GADMA2 enables approximation of selection rates and dominance coefficients for automatically constructed demographic models.

Population Size Dynamics

GADMA2 provides additional flexibility for population size estimation during model construction. Previously, demographic parameters such as functions of population size changes were estimated within a fixed set of three possible dynamics: constant, linear, or exponential change. Now, the list of available population size dynamics in GADMA2 can be appointed to any subset of three basic functions. Thus, for example, linear size change can be excluded from the demographic inference if only constant and exponential dynamics are applicable, for example, for *momi2* engine.

Inbreeding coefficients

Since the publication of the first version of GADMA, the supported simulation engines were also upgraded. GADMA2 follows these changes and includes inference of inbreeding coefficients that were implemented in ∂a∂i (Blischak et al., 2020). Using this new feature included in ∂a∂i, we demonstrate that GADMA2 provides better and more stable results for inference of the demographic models obtained from data for the puma and cabbage reported by Blischak et al. (2020) (Figure 3).

2.3 Data formats

Another improvement of ∂a∂i and *moments* is the ability to build an AFS dataset directly from a VCF file. Before this feature was implemented, this had to be done either manually or using another software like *easySFS* (`https://github.com/isaacovercast/easySFS`). GADMA2 is able to read data directly from a VCF file and to downsize, exclude populations from, or build a folded AFS automatically. Such feature allows wider and more convenient usage of GADMA2.

2.4 A new simulation engines

In addition to ∂a∂i and *moments*, GADMA2 now includes two new simulation engines: *momi2* (Kamm et al., 2020) and *momentsLD* (Ragsdale and Gravel, 2019, 2020). Thus, four engines are provided in the common interface of GADMA2. The ∂a∂i and *moments* engines are based on the Wright-Fisher model which assumes generations to be non-overlapping and both use allele frequency spectrum statistics for demographic inference.

In contrast to the Wright-Fisher model, the Moran model used in *momi2* (Kamm et al., 2020) reflects reality better. *Momi2* also uses AFS data, but it does not support continuous migration and linear change in population size. Also, it is computationally faster than ∂a∂i and *moments* and can handle up to ten populations. Therefore, *momi2* was included as a simulation engine in GADMA2.

Even though the allele frequency spectrum is one of the most popular statistics for demographic inference, it has several limitations on how informative it can be (Myers et al., 2008). The software *moments* has a submodule *momentsLD* that is dedicated to demographic inference using linkage disequilibrium (LD) statistics. A new simulation engine using *momentsLD* was integrated GADMA2. It is the first simulation engine that does not use AFS-based statistics. Overall, GADMA2 now provides a choice of four engines and we encourage the community to extend this list.

We compared four engines supported by GADMA2 on the simulated dataset of two orang-utan species. The original demographic model includes migrations, however, the *momi2* engine does not support continuous migrations. Thus, two demographic histories were analysed: 1) without migrations; 2) with migrations. The simulated parameter values and their estimations inferred by engines in GADMA2 are presented in the Table A4 for model 1 and in Table A5 for model 2. The predicted parameters of both demographic histories turned out to be very similar between ∂a∂i, *moments* and *momi2* engines that are based on the allele count statistics. *MomentsLD* provided better estimations for demographic model without migrations (model 1). Discrepancies between predicted and simulated parameter values could be explained by the fact that model 1 is oversimplified and lacks migration events. Estimations provided by all engines were close to the simulated parameter values for model with migrations (model 2).

Moreover, performance of *momi2* engine was analysed for demographic histories with different number of pulse migrations using the same dataset. The time interval between present and time of divergence was divided in even parts and pulse migrations with equal rate were integrated between them. Parameters of four *momi2* demographic models were inferred with 0, 1, 3 and 7 pulse migrations. The results of parameter estimations are presented in Table A6. Inferred pulse migration rates differs significantly from continuous rates used in simulation but they became more accurate with increased number of pulse events. Other parameters also converge to the simulated parameter values. Continuous migration is not supported in *momi2* engine but to some degree could be replaced by several pulse migration events.

2.5 A new engine for demographic history representation

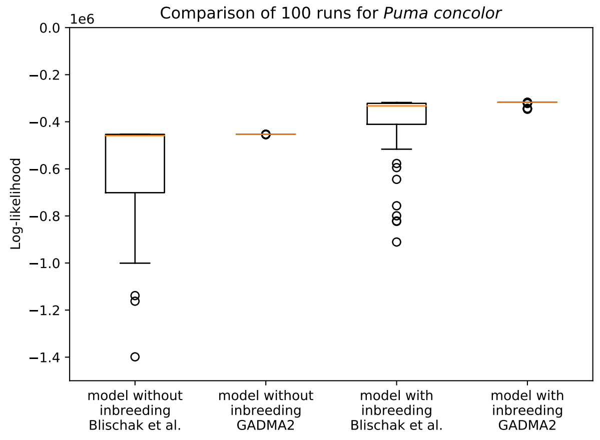
During demographic model inference, GADMA provides different text and visual representations of the current best demographic history, for example, generated Python code for all available simulation engines or picture with visualised demographic history. Recently, a new Python package named *demes* (Gower et al., 2022) appeared to allow standard human-readable descriptions of demographic histories. GADMA2 includes *demes* as an engine to generate native descriptions and plots of demographic histories, which was only possible to do before using the *moments* or *momi2* engine. Figures 4 and 5 show the examples of visual representations of demographic history using *demes*.

2.6 Availability

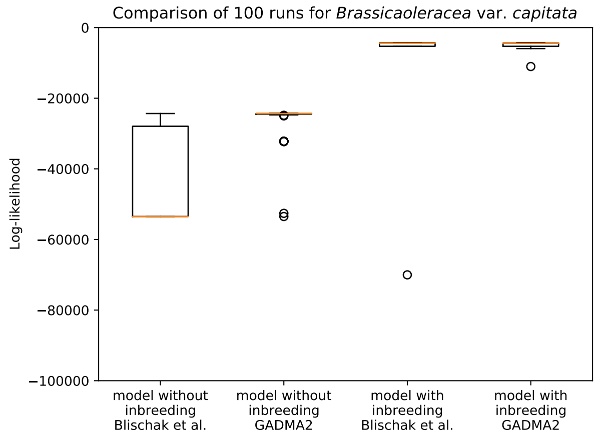
GADMA2 is freely available from GitHub via the link `https://github.com/ctlab/GADMA` and can be easily installed via Pip or BioConda. Detailed documentation is located on the website `https://gadma.readthedocs.io` and includes a user-manual, ready-to-use examples, and a section about Application Programming Interface (API). API enables an opportunity to use GADMA2 as a Python package and allows its optimization algorithms to be applied for any general optimization problem. An example of such usage is demonstrated for Rosenbrook function optimization, which is provided in the documentation.

2.7 Usage case: inference of inbreeding coefficients

We performed demographic inference with GADMA2 using the data of American pumas (*Puma concolor*) and domesticated cabbage (*Brassica oleracea* var. *capitata*) from Blischak et al. (2020).

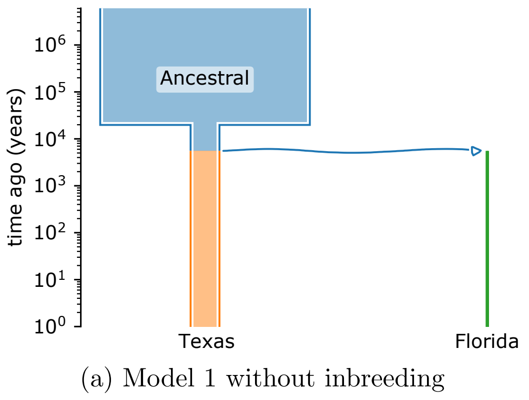


(a)



(b)

Figure 3: Boxplots for results of log-likelihoods obtained from 100 runs for demographic history inference of (a) American Puma, (b) domesticated cabbage. Two models were used from Blischak et al. (2020): with and without inbreeding. Results from GADMA2 were compared to the results of 100 runs from the original paper by Blischak et al. (2020) that were received by optimization techniques implemented in ∂a∂i. GADMA2 provided more accurate and stable solutions.



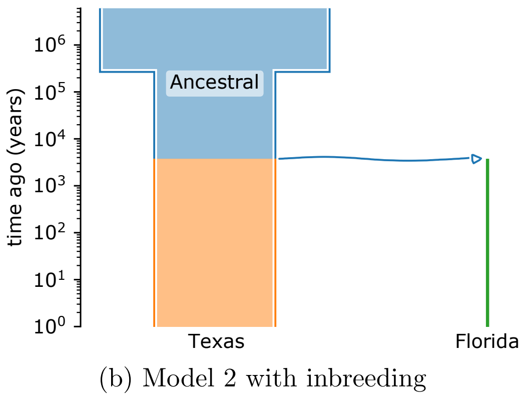


Figure 4: Demographic histories for Texas and Florida populations of American puma inferred with GADMA2. Figures were generated with the *demes* package (Gower et al., 2022). Time is presented in log scale.

Table 2: Maximum likelihood parameters inferred from the demographic models for the Texas and Florida populations of American puma.

	Model 1 Blischak et al. (2020)	Model 1 GADMA2	Model 2 Blischak et al. (2020)	Model 2 GADMA2
Number of parameters	4	4	6	6
Log-likelihood	−453,003.05	−452,475.41	−318,058.08	**−316,109.44**
Population size (95% CI)				
N_A	120,000 (92,400 – 157,000)	118,141 (98,447 – 141,775)	130,000 (129,000 – 132,000)	133,934 (130,623 – 137,329)
N_{TX}	23,700 (3,490 – 161,000)	16,261 (992 – 266,471)	70,800 (63,300 – 79,200)	70,688 (68,451 – 72,998)
N_{FL}	1,210 (118 – 12,500)	821 (120 – 5,624)	1,600 (128 – 19,100)	769 (0 – 1,971,771)
Time in years (95% CI)				
T_1	26,800 (504 – 1,420,000)	14,250 (104 – 1,953,825)	247,000 (169,000 – 359,000)	263,061 (242,533 – 285,326)
T_2	8,230 (784 – 86,500)	5,548 (790 – 38,982)	7,820 (650 – 94,200)	3,764 (2 – 8,473,716)
Inbreeding coefficients (95% CI)				
F_{TX}	NA	NA	0.440 (0.408 – 0.474)	0.453 (0.393 – 0.521)
F_{FL}	NA	NA	0.607 (0.588 – 0.626)	0.628 (0.576 – 0.685)

N_A: size of ancestral population; N_{TX}: size of ancestral population after growth and size of Texas population; N_{FL}: size of Florida population after divergence; T_1: time of epoch between ancestral population size growth and split event; T_2: time of divergence; F_{TX}: inbreeding coefficient for Texas population; F_{FL}: inbreeding coefficient for Florida population. Best log likelihood value.

For each dataset two demographic models were inferred: 1) a model from the original paper without inbreeding; 2) a model from the original paper with inbreeding included as a function. Each demographic inference was run 100 times and the model with highest likelihood value was considered the best.

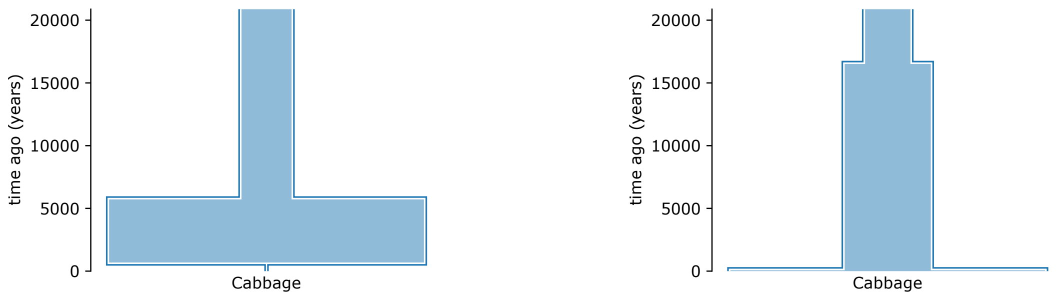
In order to compare the performance of GADMA2 with ∂a∂i, model 1 and model 2 were inferred with GADMA2 with the same parameter bounds as were used by Blischak et al. (2020). GADMA2 provided better and more stable results within 100 runs for both datasets (Figure 3). However, several model parameters received values that were close to their upper or lower bounds (not presented). In order to avoid this limitation, we performed another inference with wider bounds for parameter values and observed more valid demographic parameters. Final values of the parameters are presented in Table 2 for American pumas and in Table 3 for domesticated cabbage. Two result models were compared with the likelihood ratio test (Coffman et al., 2016) to investigate which model was more likely.

2.7.1 American puma demographic history

The best demographic histories obtained with GADMA2 had better values of log-likelihood ($-452{,}475.41$ vs $-453{,}003.05$ for model 1 and $-316{,}109.44$ vs $-318{,}058.08$ for model 2) than those reported in Blischak et al. (2020). Similar values of population sizes were obtained except for the size of Florida population, which was estimated to be 800 individuals compared to the $1{,}200 - 1{,}600$ individuals estimated by Blischak et al. (2020). Time of divergence was estimated as $4{,}000 - 5{,}500$ years ago. Inbreeding coefficients for model 2 were reported to be a little higher than for the same model in Blischak et al. (2020): 0.453 for the Texas population and 0.628 for the Florida population. The Godambe-adjusted likelihood ratio statistic is 2568.59 (P value = 0.0; Coffman et al. (2016)), indicating that the model with inbreeding better describes data.

2.7.2 Domesticated cabbage demographic history

The best demographic histories obtained with GADMA2 for one population of domesticated cabbage had better values of log-likelihood (-24137.13 vs -24330.40 for model 1 and -4267.14 vs



(a) Model 1 without inbreeding (b) Model 2 with inbreeding

Figure 5: Demographic histories for a single population of domesticated cabbage inferred with GADMA2. Figures were generated with the *demes* package. In both models, time of the most recent epoch was estimated to be small.

Table 3: Maximum likelihood parameters inferred from the demographic models for the domesticated cabbage population.

	Model 1 Blischak et al. (2020)	Model 1 GADMA2	Model 2 Blischak et al. (2020)	Model 2 GADMA2
Number of parameters	5	5	6	6
Log-likelihood	−24,330.40	−24,137.13	−4,281.14	**−4,267.14**
Population size (95% CI)				
N_A	19,100 (18,500 – 19,800)	19,163 (16,751 – 21,923)	17,500 (16,900 – 18,100)	17,496 (16,425 – 18,636)
N_1	123,000 (80,400 – 190,000)	111,740 (24,388 – 511,961)	31,600 (28,900 – 34,700)	31,768 (28,658 – 35,216)
N_2	592 (547 – 641)	6 (2×10^{-13} – 2×10^{14})	215,000 (4,910 – 9,370,000)	174,961,828 (6,280 – 5×10^{12})
Time in years (95% CI)				
T_1	5,870 (5,200 – 6,620)	5,905 (1,421 – 24,535)	16,600 (12,900 – 21,200)	16,440 (11,252 – 24,021)
T_2	38.3 (32.5 – 45.1)	0.383 (1.4×10^{-15} – 10^{14})	322 (94.2 – 1,097)	258 (141 – 474)
Inbreeding coefficient (95% CI)				
F	NA	NA	0.578 (0.557 – 0.599)	0.578 (0.556 – 0.599)

N_A: size of ancestral population; N_1: size of population during the first epoch; N_2: size of population during the first epoch; T_1: time of the first epoch; T_2: time of the second epoch; F: inbreeding coefficient.

−4281.14 for model 2) than those reported in the original paper by Blischak et al. (2020). Similar values for the population sizes in the first and second epochs were obtained. However, the population size for the most recent epoch was underestimated (6 vs 592 individuals) for model 1 without inbreeding and overestimated (174,960,000 vs 215,000 individuals) for model 2 with inbreeding. The time duration of the epoch was also smaller for both models than estimated previously by Blischak et al. (2020). In the case of model 1 the time parameter was very close to zero. The likelihood ratio test showed that the model with inbreeding describes the data better than the model without inbreeding (LRT statistic = 127.10, P value = 0.0; Coffman et al. (2016)).

3 Discussion

GADMA2 is an extension of the initial version of GADMA. It has an updated genetic algorithm, more flexible specification of the demographic model, and a greater number of simulation engines. Based on our tests of two empirical datasets, GADMA2 provides more accurate and stable performance.

Hyperparameter optimization with SMAC was used to improve a genetic algorithm and provided several alternative configurations of hyperparameters. We note that SMAC failed to find a better solution than the default one for a configuration of all hyperparameters including two discrete. With fixed discrete hyperparameters new solutions were investigated. This supports the assumption that the presence of discrete hyperparameters can negatively influence the optimization and may require a greater number of iterations. Two result configurations showed better performance than the initial genetic algorithm. The first one obtained on round 2 has the same values of discrete hyperparameters as the default and demonstrated the best convergence on the first iterations for three simulation engines. Another configuration from round 6 has slower convergence on the first iterations but was better on the last iterations and has a more stable result over several runs in case of *moments* engine. The round 2 configuration was chosen as an universal updated genetic algorithm for GADMA2. However, we note that the solution from round 6 should be also considered as a possible enhancement in case of *moments* simulation engine for the datasets with slow convergence.

We note that the performance of the genetic algorithm before and after hyperparameter optimization was validated only for ∂a∂i, *moments* and *momi2* engines. In case of controversial performance for other engines, we recommend that hyperparameter optimization be run for each engine separately. This area related to performance requires more research.

Two new simulation engines, *momi2* and *momentsLD*, were incorporated into GADMA2. *Momi2* is based on the Moran generations model that reflects reality better than the Wright-Fisher model used in ∂a∂i and *moments*. Unfortunately, the method from *momi2* has some limitations including absence of continuous migrations and linear population size growth. *MomentsLD* is the first engine in GADMA2 that does not use allele frequency spectrum data for demographic inference but linkage disequilibrium statistics instead.

Accuracy of GADMA2 engines was analysed on simulated dataset. All engines displayed consistent performance and estimations close to simulated values. We also demonstrated that continuous migration for *momi2* engine could be replaced with several pulse migrations. However, this approach is limited due to increase in computation time for larger number of pulse migrations.

Moreover, the new package *demes* was incorporated as an engine into GADMA2 to provide text and better visual representations of the demographic history models. Collectively, these new engines provide valuable extensions of GADMA, allowing the inference of complex demographic histories for up to 10 populations.

GADMA2 allows more flexible specification of the demographic model enabling inference of migrations (gene flow), inbreeding coefficients, selection and dominance rates. We have demonstrated better performance of GADMA2 on real datasets with inbreeding.

The best demographic history inferred for two populations of American pumas included inbreeding. Our results demonstrated very broad confidence intervals for population size of the Florida puma population and time of divergence in comparison to other parameters and models. The demographic models of domesticated cabbage population introduced in Blischak et al. (2020) originally were three epoch models. The time of the third epoch was inferred to be very small both for models with and without inbreeding. Thus, the wide confidence intervals for population size during the third epoch can be explained by the fact that very recent events are difficult to investigate with ∂a∂i. We only tested the demographic models that were presented in Blischak et al. (2020) but further models can be built based on our results.

GADMA2 extends the existing version of GADMA that has already shown itself as a powerful and efficient software for inference of complex demographic histories from genetic data. GADMA2 can be further improved through integration of new simulation engines, new algorithms for optimization, and automatic model construction.

4 Methods and materials

4.1 Data availability

Several datasets were used in this work. Datasets used for hyperparameter optimization were taken from the Python package `deminf_data v1.0.0` (Figure 6) available on GitHub via the link: `https://github.com/noscode/demographic_inference_data`. Each dataset in this package includes: a) the allele frequency spectrum data; b) model of the demographic history; and c) bounds of the model parameters. The package `deminf_data` contains different datasets with both real and simulated AFS data. Simulations were performed with the *moments* (Jouganous et al., 2017) software. The full description of the AFS data and demographic model parameters are available in the repository on GitHub.

We run tests for engines of GADMA2 on simulated dataset for two orang-utan species available in *stdpopsim* library (Adrion et al., 2019). For simulation purpose we used previously described scenario of demographic history of two orang-utan species Bornean (*Pongo pygmaeus*) and Sumatran (*Pongo abelii*) (Locke et al., 2011). Specifically it is an isolation-with-migration model that describes the ancestral population split followed by exponential growth in Sumatran and exponential decay in Bornean population. We simulated 23 autosomal chromosomes with total length of 2.87 Gbp using *msprime* engine (Kelleher et al., 2016) in *stdpopsim* (Adrion et al., 2019). Mutation rate used in simulation was equal to $1.5 \cdot 10^{-8}$ per site per generation (Nater et al., 2017). Averaged recombination rates for each chromosome were taken from the *Pongo abelii* recombination map inferred in Nater et al. (2017).

Datasets for the demographic inference with inbreeding were taken from the original paper Blischak et al. (2020). The 11×5 AFS data for two populations of the American puma (*Puma concolor*) was constructed on the basis of Ochoa et al. (2019). The AFS data for 45 individuals of domesticated cabbage (*Brassica oleracea*) were obtained from publicly available resequencing data (Cheng et al., 2016a,b). Both allele frequency spectra were folded due to lack of information about ancestral alleles. Datasets are presented in the repository of the original article and are available via the following link: `https://github.com/pblischak/inbreeding-sfs`.

The scripts and results of hyperparameter optimization experiments with SMAC were saved in the repository and are available via the link: `https://github.com/noscode/HPO_results_GADMA`. The results of GADMA runs for different hyperparameter configurations were stored as an

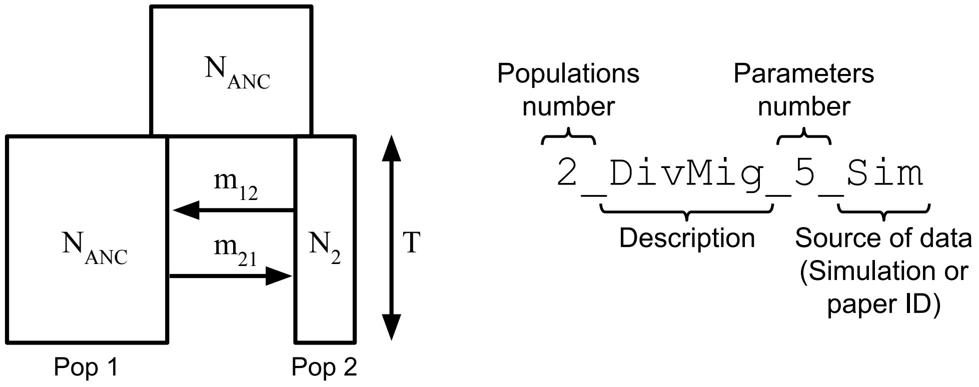


Figure 6: Structure of dataset name from `deminf_data v1.0.0` package.

archive which is available via the link: `https://ctlab.itmo.ru/files/papers_files/GADMA2/comparison_on_datasets.zip`. The results of experiments about inbreeding were added to the repository with final demographic histories inferred in the original paper of GADMA and are located via the link: `https://bitbucket.org/noscode/gadma_results`.

4.2 Hyperparameter optimization

The performance of any algorithm depends on the values of its hyperparameters. The learning rate of neural network is one of the classical examples of such dependence. One of the most popular frameworks for hyperparameters search is *Bayesian optimization*. The problem of finding hyperparameters that provide best convergence not on one but on several instances of data is called *algorithm configuration problem*. Algorithm configuration problem can be stated as follows: given a target algorithm A, a set of problem instances I and a *cost function* (metric) c, seek for the best set of hyperparameters for A on I in regard to c. Usually, cost function is based on time required to solve the problem or on quality of solution that was achieved within a given budget. SMAC (Hutter et al., 2011, Lindauer et al., 2021) is the software that implements Bayesian optimization for solving the algorithm configuration problem.

SMAC is based on Bayesian optimization, which is a model-based algorithm. It uses a *surrogate model* to estimate objective function and *acquisition function* to find new promising points for function evaluation. SMAC uses Random forest as a surrogate model for cost function approximation and expected improvement as an acquisition function to make predictions of promising hyperparameter configuration on each iteration. In contrast to classical Bayesian optimization SMAC can handle several problem instances and therefore has some modifications. First of all, the objective function that is modelled with Random forest is a mean value of cost functions on instances. Another extension is the procedure of intensification. It is a mechanism that governs how to compare the new configuration with the existing best configuration (incumbent). Random Online Aggressive Racing (ROAR) implemented in SMAC considers a new configuration as a new incumbent if it is better than the current incumbent on the set of problem instances and random seed pairs. This set is extended each time when a new configuration is turned out to be worse than the incumbent. Such an algorithm falls into the fact that this set is always growing and more and more comparisons are required to beat the current best configuration.

In summary, SMAC is an iterative algorithm that keeps the best found configuration of hyperparameters. On each iteration new promising configuration is chosen and is compared to the current incumbent within the procedure of intensification. The comparison is managed by the value of cost function averaged over problem instances, as the result the best by average algorithm performance hyperparameter configuration is provided.

The optimization method such as genetic algorithm has several hyperparameters and their number varies within specific implementations. The genetic algorithm presented in GADMA maintains a set of problem solutions that is called generation. Each solution is presented as an array of values for objective function parameters, i.e. parameters of the demographic history. The initial generation is formed by the *initial design* procedure: a set of random solutions is created. The size of this set is determined by the value of hyperparameter `n_init_const`: the number of solutions in the initial generation is equal to the number of target parameters multiplied by `n_init_const`. Size of each generation in the genetic algorithm is equal to the value of the `gen_size` hyperparameter. New generation is constructed iteratively with the help of mutation, crossover and selection of best by the value of likelihood models. The fractions of most adapted,

Table 4: Short descriptions of GADMA genetic algorithm (GA) hyperparameters.

Hyperparameter ID	Hyperparameter description
`gen_size`	Number of solutions in generation of genetic algorithm
`n_init_const`	Constant that determines number of random solutions that are created during initial design at the beginning of GA
`p_elitism`	Fraction of the best solutions that are taken to new generation
`p_mutation`	Fraction of mutated solutions in a new generation
`p_crossover`	Fraction of crossed solutions in a new generation
`p_random`	Fraction of random solutions in a new generation
`mutation_strength`	Initial parameter change probability for mutation
`const_mutation_strength`	Constant to change a `mutation_strength` during genetic algorithm according to one-fifth rule
`mutation_rate`	Initial rate of a parameter change during mutation
`const_mutation_rate`	Constant to change the `mutation_rate` during genetic algorithm according to one-fifth rule

mutated, crossed and random models that form a new generation are determined by `p_elitism`, `p_mutation`, `p_crossover`, `p_random` hyperparameters correspondingly. The special case is the mutation process: it is determined by `mutation_strength` and `mutation_rate` that define how many parameters of the model and how strong their values will change. Each of these two hyperparameters are changed during the genetic algorithm performance according to one-fifth rule: the closer to the optimum we are the smaller changes during the mutation process are. The constants of one-fifth rule (`const_mutation_strength`, `const_mutation_rate`) are additional two hyperparameters of the genetic algorithm. In total we highlight 10 hyperparameters for the genetic algorithm implemented in GADMA: two have integer values and eight are continuous.The short descriptions of each hyperparameter are presented in Table 4.

The initial values of hyperparameters in the first version of GADMA were obtained manually within the demographic inference of two populations of modern humans for the model and data from Gutenkunst et al. (2009) (`2_YRI_CEU_6_Gut` dataset). Their values are presented in Table 5.

We used SMAC to investigate hyperparameter values of the genetic algorithm in GADMA. As a result we have performed six rounds of SMAC optimization for different configurations of hyperparameters (Table 1). The domains of hyperparameter values that were used are presented in Table 5. Each round was running for two weeks, in parallel on 10 processes. In order to achieve the valid comparison within the SMAC framework GADMA runs were stopped after a fixed number of likelihood evaluations. We took 200 $\times$ number of parameters as the stop criteria for runs of the genetic algorithm in SMAC. Such a number of evaluations was chosen to be a trade-off between speed and accuracy: according to the convergence plots at this point convergence of default genetic algorithm optimization was slowing down and was very close to the plateau walk (Figure A1, Figure A2).

Four datasets were chosen as train problem instances for SMAC. The result configurations were tested and compared to the default genetic algorithm on train and on additional six test datasets. All used datasets were taken from Python package `deminf_data v1.0.0`. Each dataset presented in Table A1, Table A2 and Table A3 had a structured name (Figure 6) that is the sequence of

Table 5: Default values and domains used for optimization of GADMA genetic algorithm hyperparameters. The first two hyperparameters `gen_size` and `n_init_const` are integers and have discrete domains. Other eight hyperparameters are continuous.

Hyperparameter ID	Default value	Domain
`gen_size`	10	$\{10, 50, 100\}$
`n_init_const`	10	$\{5, 10, 20\}$
`p_elitism`	0.2	$[0, 1]$
`p_mutation`	0.3	$[0, 1]$
`p_crossover`	0.3	$[0, 1]$
`p_random`	0.2	$[0, 1]$
`mutation_strength`	0.2	$[0, 1]$
`const_mutation_strength`	1.01	$[1, 2]$
`mutation_rate`	0.2	$[0, 1]$
`const_mutation_rate`	1.02	$[1, 2]$

a) population number, b) short description of the demographic model, c) number of parameters and d) information about the source of AFS data.

Each of four train datasets had demographic model and data for two populations: three simulated AFS data (`2_BotDivMig_8_Sim`, `2_DivMig_5_Sim`, `2_ExpDivNoMig_5_Sim`) and one real data (`2_YRI_CEU_6_Gut`) for modern human populations from Gutenkunst et al. (2009). Thus, datasets require similar resources for likelihood evaluations and are well balanced for optimization with SMAC.

Test datasets were chosen to be more diverse: two datasets (`1_Bot_4_Sim`, `1_AraTha_4_Hub`) for one population with simulated and real data from Huber et al. (2018), two datasets (`2_ButAll_3_McC`, `2_ButSynB2_5_McC`) for two populations of butterflies from McCoy et al. (2014); one simulated dataset for three populations (`3_DivMig_8_Sim`); and one dataset (`2_YRI_CEU_struct_11_Nos`) with AFS data for two populations of modern human from Gutenkunst et al. (2009) and unsupervised demographic history with structure (2, 1) that was observed in the original paper of GADMA Noskova et al. (2020).

During the first run of SMAC all ten hyperparameters (Table 1) were optimised. However, SMAC failed to find a better configuration than the default one. Two discrete hyperparameters (`gen_size` and `n_init_const`) were excluded from the configuration and more rounds were launched in order to perform grid search.

Discrete hyperparameters `gen_size` and `n_init_const` were fixed to default values (10 and 10 correspondingly) for the second round of SMAC optimization (Table 1). For the third and fourth rounds we tested two alternative values (5 and 20) of initial design constant (`n_init_const`). Then the number of demographic models on each generation of genetic algorithm (`gen_size`) was increased up to 50 and constant for initial design (`n_init_const`) was tested for values of 10 and 20. The value of `n_init_const` equal to 5 was excluded from the experiments as it provides a small number of solutions for the first generation that should be of size 50.

SMAC compares configurations based on mean cost score evaluated over all provided training datasets. This score was independently received from 128 runs per dataset and presented in Table A1 for *moments* engine. Incumbents for rounds 2, 3 and 4 where `gen_size` was fixed to the value of 10 showed better SMAC scores than an initial configuration. The round 3 configuration showed the best score among them. The final incumbents for rounds 5 and 6 with `gen_size` fixed to the value of 50 showed a worse SMAC score than the default configuration (Table A1). Configurations from rounds 4, 5 and 6 provided high mean log-likelihoods on three of four training datasets but failed to perform accurate convergence on the `2_ExpDivNoMig_5_Sim` dataset. We also note that a genetic algorithm with hyperparameters from round 6 turned out to be more effective than the incumbent from round 5 on all training and test datasets according to the mean costs (Table A1). The round 2 configuration showed best mean scores on two training datasets (`2_DivMig_5_Sim`, `2_ExpDivNoMig_5_Sim`), and the round 6 configuration turned out to be the best on the other two training datasets (`2_BotDivMig_8_Sim`, `2_YRI_CEU_6_Gut`).

Further, final hyperparameter configurations were validated on six test datasets with the mean costs associated with these datasets presented in Table A1. As in case of the training datasets, two configurations from round 2 and round 6 showed the best performance on half (3 of 6) of the test datasets. Thus, these two configurations were compared in terms of convergence on a greater number of iterations. The convergence plots of genetic algorithms with the initial configuration and configurations obtained on 2 and 6 rounds are presented in Figure A1 for the training datasets and in Figure A2 for the test datasets. Mean scores of the final configurations were also evaluated for the ∂a∂i (Table A2) and *momi2* (Table A3) engines.

4.3 Performance test of GADMA2 engines

Four GADMA2 engines were compared on simulated dataset of two orang-utan species. Two demographic histories were used: 1) isolation with the ancestral population split followed by exponential growth in Sumatran and exponential decay in Bornean population 2) isolation-with-migration with the ancestral population split followed by exponential growth in Sumatran and exponential decay in Bornean population. Mutation and recombination rates were taken the same as were used in the simulation. Performance of all four engines was compared for model 1, however, *momi2* engine was not validated for model 2 as *momi2* does not support continuous migrations. We ran GADMA 8 times for each engine and model.

In order to overcome *momi2* limitation with continuous migrations we validated engine on additional demographic scenarios with pulse migrations. Different number of pulse migrations with equal rate were uniformly integrated within the epoch between present time and species divergence time. Four demographic models were tested: 1) without pulse migrations, 2) with 1 pulse migration, 3) with 3 pulse migrations, and 4) with 7 pulse migrations.

4.4 Inbreeding

Mutation rates and sequence lengths used for parameter translation from genetic units were taken the same as in Blischak et al. (2020). Demographic parameters for *Puma concolor* models were translated from the genetic to real units using a mutation rate of $\mu = 2.2 \times 10^{-9}$, a generation time of 3 years, and a sequence length of 2,564,692,624 bp (Ochoa et al., 2019). In the case of *Brassica oleracea* var. *capitata* population demographic parameters were translated using mutation rate of $\mu = 1.5 \times 10^{-8}$, a generation time of 1 year, and a sequence length of 411,560,319 bp.

Confidence intervals reported in Table 2 and Table 3 were estimated on 100 bootstrapped AFS data from the original paper using the Godambe information matrix with step size equal to $\epsilon = 10^{-2}$

(Coffman et al., 2016). The scripts and data used for CI evaluation were taken from the repository of Blischak et al. (2020) article: `https://github.com/pblischak/inbreeding-sfs`.

5 Funding

E. Noskova, P. Dobrynin and V. Ulyantsev were financially supported by Ministry of Science and Higher Education of the Russian Federation (Priority 2030 Federal Academic Leadership Program). E. Noskova was also supported by Systems Biology Program by Skoltech.

References

Ryan N Gutenkunst, Ryan D Hernandez, Scott H Williamson, and Carlos D Bustamante. Inferring the joint demographic history of multiple populations from multidimensional snp frequency data. *PLoS genetics*, 5(10):e1000695, 2009.

Julien Jouganous, Will Long, Aaron P Ragsdale, and Simon Gravel. Inferring the joint demographic history of multiple populations: beyond the diffusion approximation. *Genetics*, 206(3):1549–1567, 2017.

Matthias Steinrücken, Jack Kamm, Jeffrey P Spence, and Yun S Song. Inference of complex population histories using whole-genome sequences from multiple populations. *Proceedings of the National Academy of Sciences*, 116(34):17115–17120, 2019.

Jack Kamm, Jonathan Terhorst, Richard Durbin, and Yun S Song. Efficiently inferring the demographic history of many populations with allele count data. *Journal of the American Statistical Association*, 115(531):1472–1487, 2020.

Laurent Excofffier, Nina Marchi, David Alexander Marques, Remi Matthey-Doret, Alexandre Gouy, and Vitor C Sousa. fastsimcoal2: demographic inference under complex evolutionary scenarios. *Bioinformatics*, 2021.

William S DeWitt, Kameron Decker Harris, Aaron P Ragsdale, and Kelley Harris. Nonparametric coalescent inference of mutation spectrum history and demography. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

Aaron P Ragsdale and Ryan N Gutenkunst. Inferring demographic history using two-locus statistics. *Genetics*, 206(2):1037–1048, 2017.

Aaron P Ragsdale and Simon Gravel. Models of archaic admixture and recent history from two-locus statistics. *PLoS genetics*, 15(6):e1008204, 2019.

Aaron P Ragsdale and Simon Gravel. Unbiased estimation of linkage disequilibrium from unphased data. *Molecular Biology and Evolution*, 37(3):923–932, 2020.

Ekaterina Noskova, Vladimir Ulyantsev, Klaus-Peter Koepfli, Stephen J O'Brien, and Pavel Dobrynin. Gadma: Genetic algorithm for inferring demographic history of multiple populations from allele frequency spectrum data. *GigaScience*, 9(3):giaa005, 2020.

Peiwen Xiong, C Darrin Hulsey, Carmelo Fruciano, Wai Y Wong, Alexander Nater, Andreas F Kautt, Oleg Simakov, Martin Pippel, Shigehiro Kuraku, Axel Meyer, et al. The comparative genomic landscape of adaptive radiation in crater lake cichlid fishes. *Molecular ecology*, 30(4): 955–972, 2021.

Lourdes Valdez and Guillermo D'Elía. Genetic diversity and demographic history of the shaggy soft-haired mouse abrothrix hirta (cricetidae; abrotrichini). *Frontiers in Genetics*, 12:184, 2021.

Elena A Pazhenkova and Vladimir A Lukhtanov. Genomic introgression from a distant congener in the levant fritillary butterfly, melitaea acentria. *Molecular Ecology*, 2021.

Loren Cassin-Sackett, Michael G Campana, Nancy Rotzel McInerney, Haw Chuan Lim, Natalia AS Przelomska, Bryce Masuda, R Terry Chesser, Eben H Paxton, Jeffrey T Foster, Lisa H Crampton, et al. Genetic structure and population history in two critically endangered kaua 'i honeycreepers. *Conservation Genetics*, pages 1–14, 2021.

Laura Buggiotti, Andrey A Yurchenko, Nikolay S Yudin, Christy J Vander Jagt, Nadezhda V Vorobieva, Mariya A Kusliy, Sergei K Vasiliev, Andrey N Rodionov, Oksana I Boronetskaya, Natalia A Zinovieva, et al. Demographic history, adaptation, and nrap convergent evolution at amino acid residue 100 in the world northernmost cattle from siberia. *Molecular Biology and Evolution*, 2021.

Paul D Blischak, Michael S Barker, and Ryan N Gutenkunst. Inferring the demographic history of inbred species from genome-wide snp frequency data. *Molecular biology and evolution*, 37(7): 2124–2136, 2020.

Ryan N Gutenkunst. dadi. cuda: Accelerating population genetics inference with graphics processing units. *Molecular biology and evolution*, 38(5):2177–2178, 2021.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pages 507–523. Springer, 2011.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization, 2021.

Jesus Lago, Fjo De Ridder, Peter Vrancx, and Bart De Schutter. Forecasting day-ahead electricity prices in europe: The importance of considering market integration. *Applied energy*, 211:890–903, 2018.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37 (1):388–427, 2021.

Shuang Wu, Xiaoning Song, Zhenhua Feng, and Xiaojun Wu. Nflat: Non-flat-lattice transformer for chinese named entity recognition. *arXiv preprint arXiv:2205.05832*, 2022.

Simon Myers, Charles Fefferman, and Nick Patterson. Can one learn history from the allelic spectrum? *Theoretical population biology*, 73(3):342–348, 2008.

Graham R Gower, Aaron P Ragsdale, Ryan N Gutenkunst, Matthew Hartfield, Ekaterina Noskova, Travis J Struck, Jerome Kelleher, and Kevin Thornton. Demes: a standard format for demographic models. *bioRxiv*, 2022.

Alec J Coffman, Ping Hsun Hsieh, Simon Gravel, and Ryan N Gutenkunst. Computationally efficient composite likelihood statistics for demographic inference. *Molecular biology and evolution*, 33(2):591–593, 2016.

Jeffrey R. Adrion, Christopher B. Cole, Noah Dukler, Jared G. Galloway, Ariella L. Gladstein, Graham Gower, Christopher C. Kyriazis, Aaron P. Ragsdale, Georgia Tsambos, Franz Baumdicker, Jedidiah Carlson, Reed A. Cartwright, Arun Durvasula, Bernard Y. Kim, Patrick McKenzie, Philipp W. Messer, Ekaterina Noskova, Diego Ortega-Del Vecchyo, Fernando Racimo, Travis J. Struck, Simon Gravel, Ryan N. Gutenkunst, Kirk E. Lohmeuller, Peter L. Ralph, Daniel R. Schrider, Adam Siepel, Jerome Kelleher, and Andrew D. Kern. A community-maintained standard library of population genetic models. *bioRxiv*, 2019. doi: 10.1101/2019.12.20.885129. URL `https://www.biorxiv.org/content/early/2019/12/21/2019.12.20.885129`.

Devin P Locke, LaDeana W Hillier, Wesley C Warren, Kim C Worley, Lynne V Nazareth, Donna M Muzny, Shiaw-Pyng Yang, Zhengyuan Wang, Asif T Chinwalla, Pat Minx, et al. Comparative and demographic analysis of orang-utan genomes. *Nature*, 469(7331):529–533, 2011.

Jerome Kelleher, Alison M Etheridge, and Gilean McVean. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS computational biology*, 12(5):e1004842, 2016.

Alexander Nater, Maja P Mattle-Greminger, Anton Nurcahyo, Matthew G Nowak, Marc De Manuel, Tariq Desai, Colin Groves, Marc Pybus, Tugce Bilgin Sonay, Christian Roos, et al. Morphometric, behavioral, and genomic evidence for a new orangutan species. *Current Biology*, 27(22):3487–3498, 2017.

Alexander Ochoa, David P Onorato, Robert R Fitak, Melody E Roelke-Parker, and Melanie Culver. De novo assembly and annotation from parental and f1 puma genomes of the florida panther genetic restoration program. *G3: Genes, Genomes, Genetics*, 9(11):3531–3536, 2019.

Feng Cheng, Jian Wu, Chengcheng Cai, Lixia Fu, Jianli Liang, Theo Borm, Mu Zhuang, Yangyong Zhang, Fenglan Zhang, Guusje Bonnema, and Xiaowu Wang. Genome resequencing and comparative variome analysis in a brassica rapa and brassica oleracea collection. sci data 3: 160119, 2016a.

Feng Cheng, Rifei Sun, Xilin Hou, Hongkun Zheng, Fenglan Zhang, Yangyong Zhang, Bo Liu, Jianli Liang, Mu Zhuang, Yunxia Liu, et al. Subgenome parallel selection is associated with morphotype diversification and convergent crop domestication in brassica rapa and brassica oleracea. *Nature genetics*, 48(10):1218–1224, 2016b.

Christian D Huber, Arun Durvasula, Angela M Hancock, and Kirk E Lohmueller. Gene expression drives the evolution of dominance. *Nature communications*, 9(1):1–11, 2018.

Rajiv C McCoy, Nandita R Garud, Joanna L Kelley, Carol L Boggs, and Dmitri A Petrov. Genomic inference accurately predicts the timing and severity of a recent bottleneck in a nonmodel insect population. *Molecular ecology*, 23(1):136–150, 2014.

A6 Appendix

A6.1 Hyperparameter optimization: SMAC costs and convergence plots using *moments* engine

Table A1: Mean log-likelihood values (128 runs) for final configurations of six SMAC rounds on train and test datasets. Log-likelihood was evaluated with *moments* simulation engine. Mean cost value on train datasets presented in the table was used by SMAC intensification procedure. For round 1 SMAC failed to find better configuration than the default one. Best mean values are marked bold.

Dataset	Round number					
	1 (default)	2	3	4	5	6
Mean cost on train datasets	−1,923.51	−1,860.98	**−1,798.85**	−1,843.54	−2,024.31	−1,955.98
Train datasets:						
`2_BotDivMig_8_Sim`	−2,328.19	−2,411.40	−1,967.04	−2,197.83	−2,042.13	**−1,963.56**
`2_DivMig_5_Sim`	−1,497.23	**−1,439.16**	−1,523.42	−1,453.18	−1,483.18	−1,441.97
`2_ExpDivNoMig_5_Sim`	−2,721.13	**−2,353.94**	−2,566.71	−3,137.09	−3,428.17	−3,137.09
`2_YRI_CEU_6_Gut`	−1,147.48	−1,139.41	−1,144.00	−1,138.40	−1,143.75	**−1,134.40**
Test datasets:						
`1_Bot_4_Sim`	−213.92	−193.49	−212.19	**−186.22**	−209.29	−212.42
`1_AraTha_4_Hub`	−96.12	**−93.05**	−96.45	−94.88	−96.54	−95.53
`2_ButAllA_3_McC`	−294.47	**−290.30**	−300.36	−294.22	−300.33	−293.99
`2_ButSynB2_5_McC`	−215.20	−214.78	−216.51	−214.55	−214.48	**−213.85**
`2_YRI_CEU_str_11_Nos`	−1,162.68	−1,164.41	−1,163.55	−1,165.00	−1,165.00	**−1,150.26**
`3_DivMig_8_Sim`	−11,791.97	−11,742.43	−11,797.04	−11,718.42	−11,742.41	**−11,637.73**

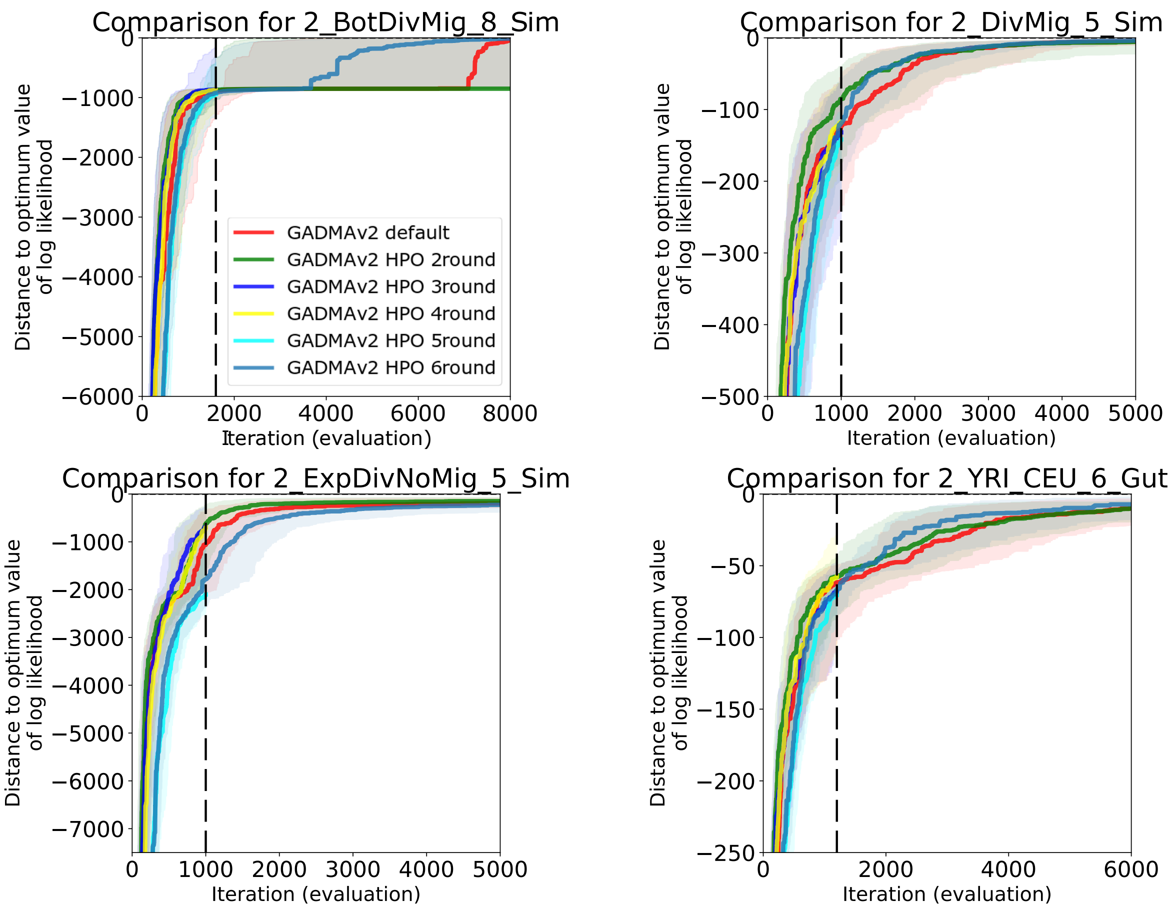


Figure A1: Convergence plots for six genetic algorithm configurations using *moments* engine on four train datasets: 1) the default genetic algorithm from the initial version of GADMA, 2)-6) configurations obtained during rounds 2-6 of hyperparameter optimization with SMAC. The abscissa presents the log-likelihood evaluation number, the ordinate refers to the distance to the optimal value of log-likelihood. Solid lines correspond to median convergence over 128 runs and shadowed areas are ranges between first (0.25) and third (0.75) quartiles. The vertical dashed black line refers to the number of evaluations used to stop a genetic algorithm in SMAC. The default configuration (red) and two configurations from round 2 (green) and round 6 (blue) were compared in terms of convergence on a greater number of iterations. The configuration from round 2 shows faster convergence on first iterations, the configuration from round 6 turns out to have better convergence at last iterations on three of four datasets.

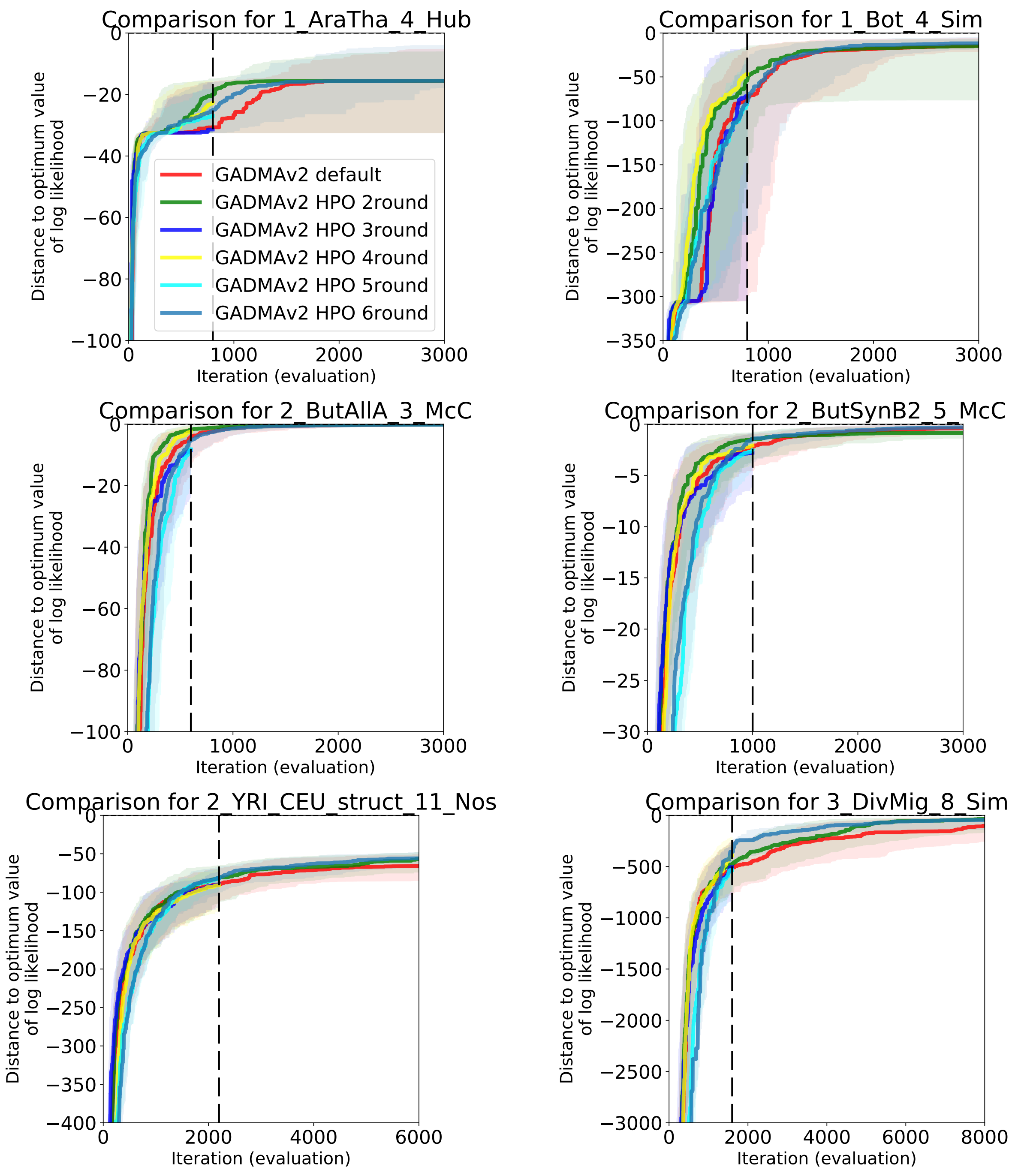


Figure A2: Convergence plots for six genetic algorithm configurations using *moments* engine on six test datasets: 1) the default genetic algorithm from the initial version of GADMA, 2)-6) configurations obtained during rounds 2-6 of hyperparameter optimization with SMAC. The default configuration (red) and two configurations from round 2 (green) and round 6 (blue) were compared in terms of convergence on a greater number of iterations. The configuration from round 2 shows faster convergence on first iterations, the configuration from round 6 turns out to have better convergence at last iterations on two of six datasets.

A6.2 Hyperparameter optimization: SMAC costs and convergence plots using ∂a∂i engine

Table A2: Mean log-likelihood values (128 runs) for final configurations of six SMAC rounds on train and test datasets using ∂a∂i simulation engine. Best mean values are marked bold. Log-likelihood values and results are similar to *moments* engine.

Dataset	Round number					
	1 (default)	2	3	4	5	6
Train datasets:						
2_BotDivMig_8_Sim	−2,392.63	−2,264.38	−2,140.06	−2,115.68	−1,998.72	**−1,908.14**
2_DivMig_5_Sim	−1,494.41	**−1,457.52**	−1,495.68	−1,461.03	−1,481.33	−1,465.06
2_ExpDivNoMig_5_Sim	−2,720.98	−2,534.38	**−2,448.77**	−2,450.89	−3,471.70	−3,329.37
2_YRI_CEU_6_Gut	−1,148.08	−1,137.60	−1,139.10	−1,135.92	−1,152.00	**−1,134.62**
Test datasets:						
1_Bot_4_Sim	−231.34	−209.55	**−170.73**	−192.93	−216.42	−227.23
1_AraTha_4_Hub	−93.27	**−92.86**	−95.80	−94.21	−95.96	−95.75
2_ButAllA_3_McC	−296.52	**−291.90**	−301.06	−297.96	−301.25	−298.50
2_ButSynB2_5_McC	−204.13	204.90	**−191.29**	−214.57	−216.21	−192.76
2_YRI_CEU_str_11_Nos	−1,173.82	−1,166.12	−1,172.60	−1,154.80	−1,153.77	**−1,148.56**
3_DivMig_8_Sim	−11,928.42	−11,766.96	−11,844.63	−11,726.69	−11,715.04	**−11,652.18**

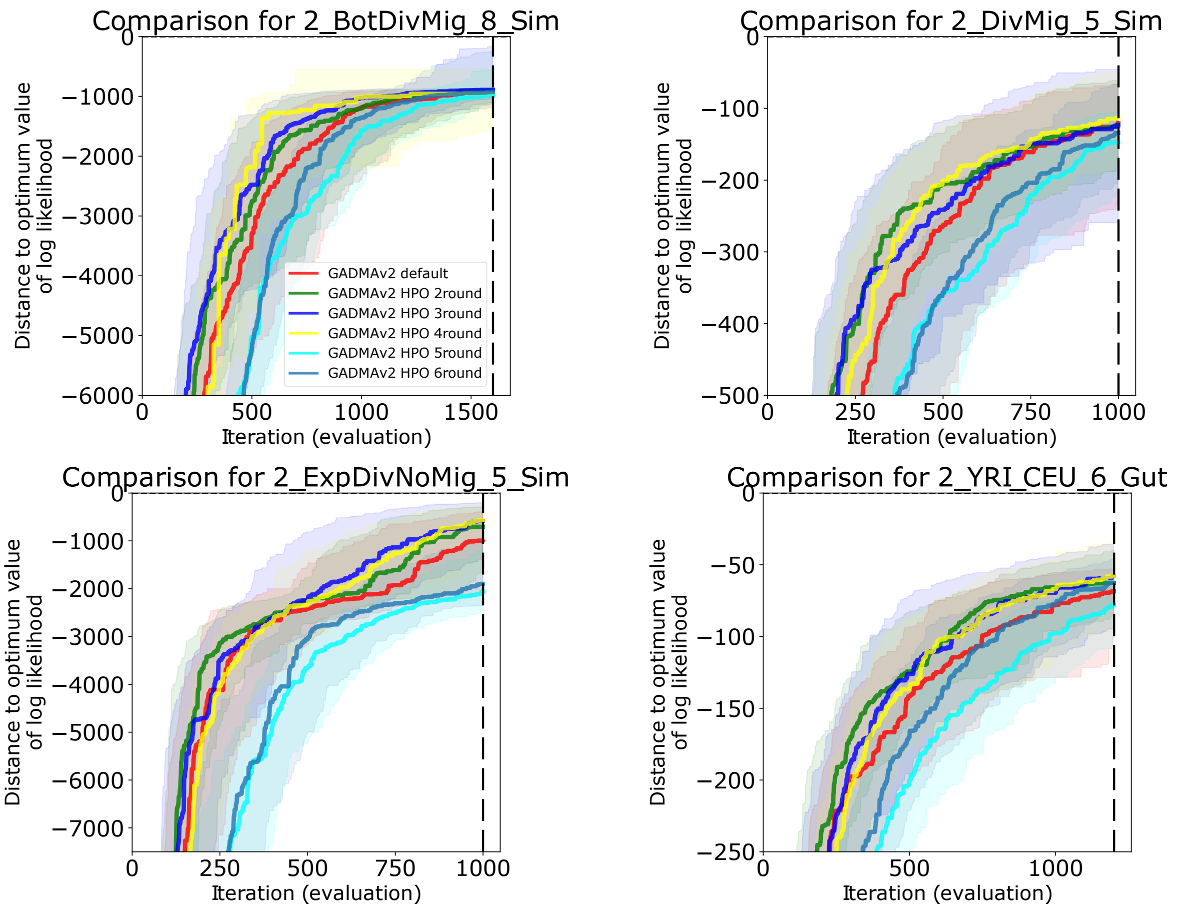


Figure A3: Convergence plots for six genetic algorithm configurations using ∂a∂i engine on four train datasets: 1) the default genetic algorithm from the initial version of GADMA, 2)-6) configurations obtained during rounds 2-6 of hyperparameter optimization with SMAC. The abscissa presents the log-likelihood evaluation number, the ordinate refers to the distance to the optimal value of log-likelihood. Solid lines correspond to median convergence over 128 runs and shadowed areas are ranges between first (0.25) and third (0.75) quartiles. The vertical dashed black line refers to the number of evaluations used to stop a genetic algorithm in SMAC.

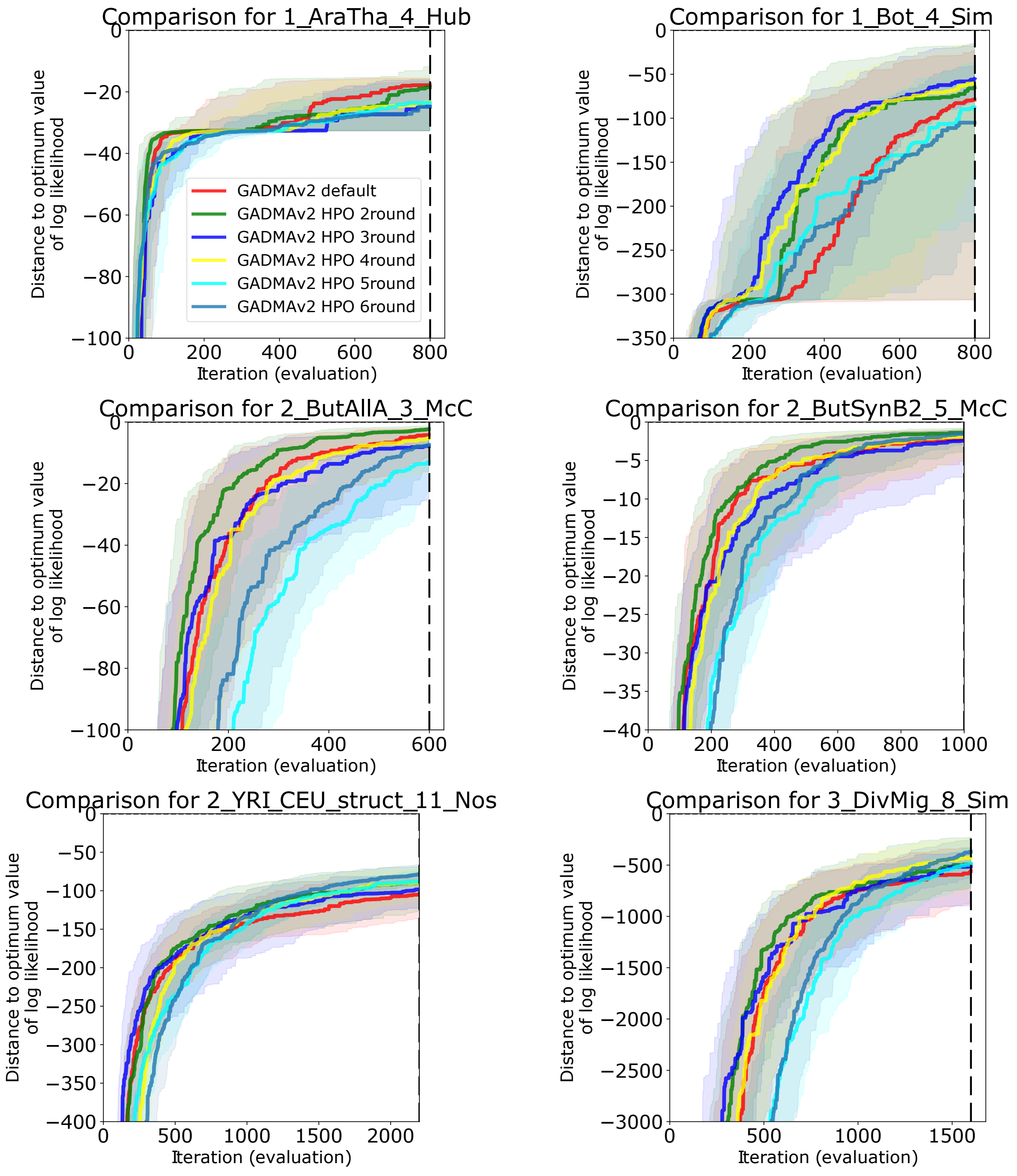


Figure A4: Convergence plots for six genetic algorithm configurations on six test datasets: 1) the default genetic algorithm from the initial version of GADMA (red colour), 2)-6) configurations obtained during rounds 2-6 of hyperparameter optimization with SMAC.

A6.3 Hyperparameter optimization: SMAC costs and convergence plots using *momi2* engine

Table A3: Mean log-likelihood values (128 runs) for final configurations of six SMAC rounds on train and test datasets using *momi2* simulation engine. Two test datasets (`2_ButAllA_3_McC`, `2_ButSynB2_5_McC`) were excluded as they lacked sequence length required for *momi2* engine. Moreover, *momi2* engine does not support continuous migrations and size of ancestral population could not be inferred implicitly as for ∂a∂i and *moments*. Thus, number of parameters in datasets for *momi2* differs from *moments* and ∂a∂i. Best mean values are marked bold.

Dataset	Par.	Round number					
	num.	1 (default)	2	3	4	5	6
Train datasets:							
`2_BotDivMig_8_Sim`	7	−362,794.00	−362,661.13	−362,708.40	**−362,436.72**	−363,522.98	−363,208.38
`2_DivMig_5_Sim`	4	−352,187.20	**−351,911.98**	−352,240.74	−352,023.01	−352,108.35	−352,059.60
`2_ExpDivNoMig_5_Sim`	6	−1,161,762.17	**−1,161,178.83**	−1,161,297.18	−1,161,422.88	−1,162,363.59	−1,161,677.16
`2_YRI_CEU_6_Gut`	6	−63,877.21	−63,866.37	−63,880.52	**−63,856.10**	−63,869.32	−63,858.89
Test datasets:							
`1_Bot_4_Sim`	5	−109,557.18	**−109,524.04**	−109,555.30	−109,524.06	−109,600.74	−109,592.11
`1_AraTha_4_Hub`	5	−228,316.33	**−228,309.85**	−228,314.93	−228,314.22	−228,352.78	−228,311.32
`2_YRI_CEU_str_11_Nos`	10	−63,882.18	−63,871.54	−63,881.70	−63,881.69	−63,880.74	**−63,860.48**
`3_DivMig_8_Sim`	6	−641,960.29	−641,757.88	−647,365.31	−642,091.09	−641,481.87	**−641,244.71**

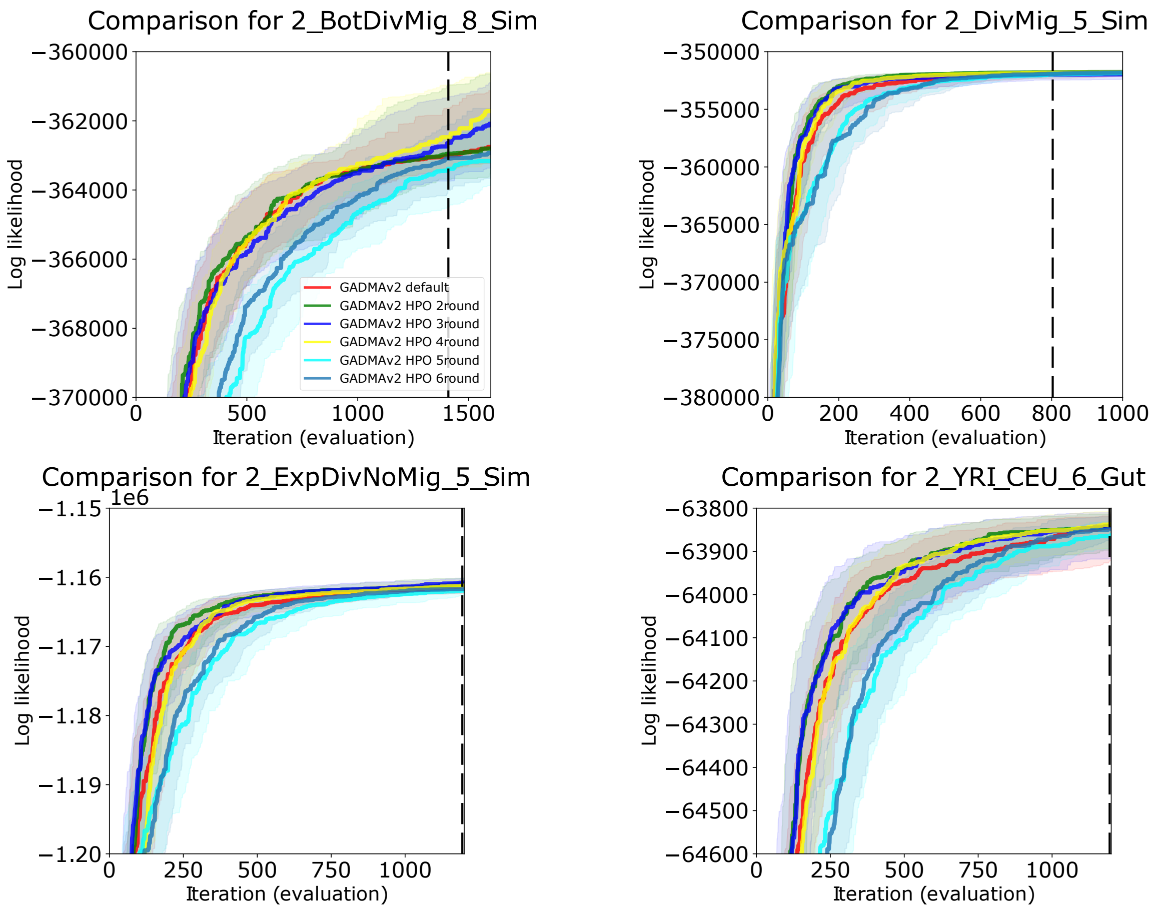


Figure A5: Convergence plots for six genetic algorithm configurations using *momi2* engine on four train datasets: 1) the default genetic algorithm from the initial version of GADMA (red colour), 2)-6) configurations obtained during rounds 2-6 of hyperparameter optimization with SMAC. The abscissa presents the log-likelihood evaluation number, the ordinate refers to the distance to the optimal value of log-likelihood. Solid lines correspond to median convergence over 128 runs and shadowed areas are ranges between first (0.25) and third (0.75) quartiles. The vertical dashed black line refers to the number of evaluations used to stop a genetic algorithm in SMAC.

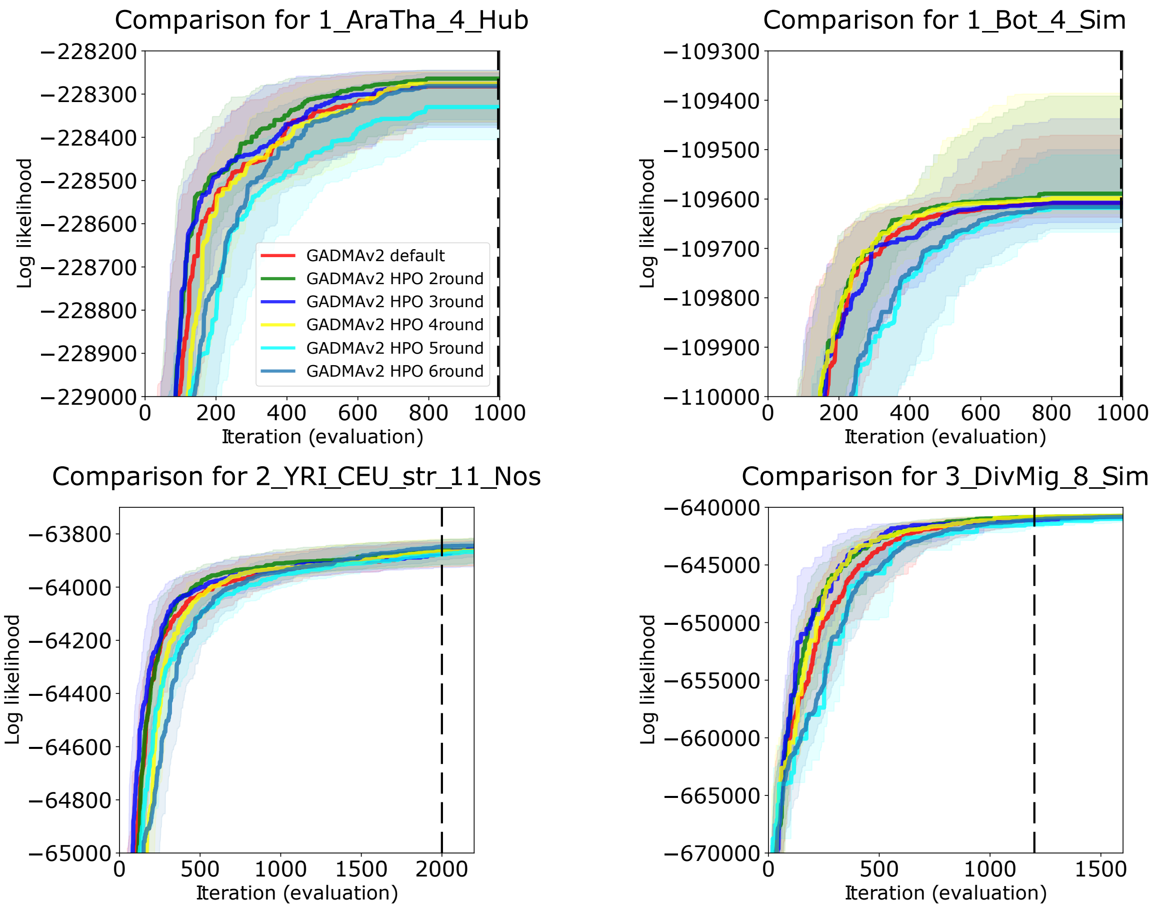


Figure A6: Convergence plots for six genetic algorithm configurations on four test datasets: 1) the default genetic algorithm from the initial version of GADMA (red colour), 2)-6) configurations obtained during rounds 2-6 of hyperparameter optimization with SMAC. Two datasets (`2_ButAllA_3_McC`, `2_ButSynB2_5_McC`) were excluded as they are not supported by *momi2* engine.

A6.4 Performance test of GADMA2 engines

Table A4: The demographic parameters of orang-utan history without migration (model 1) inferred with different engines in GADMA2. True values are the simulated parameter values that were obtained from the original paper Locke et al. (2011).

Parameter	True value	∂a∂i	*moments*	*momi2*	*momentsLD*
N_{anc}	17,934	19,834	19,776	19,331	16,390
N_{Bor_split}	10,617	6,753	6,278	6,187	7,897
N_{Sum_split}	7,317	8,516	7,753	7,719	6,666
N_{Bor}	8,805	11,039	10,886	10,663	10,427
N_{Sum}	37,661	55,733	56,415	54,184	61,577
T_{split} (gen.)	20,157	12,090	11,458	11,270	14,021

N_{anc}: size of ancestral population; N_{Bor_split}: size of *Pongo pygmaeus* at split; N_{Sum_split}: size of *Pongo abelii* at split; N_{Bor}: size of *Pongo pygmaeus* after exponential decline; N_{Sum}: size of *Pongo abelii* after exponential growth; T_{split}: time of divergence in generations.

Table A5: The demographic parameters of orang-utan history with migration (model 2) inferred with different engines in GADMA2. True values are the simulated parameter values that were obtained from the original paper Locke et al. (2011). *Momi2* engine was excluded as it does not support continuous migrations.

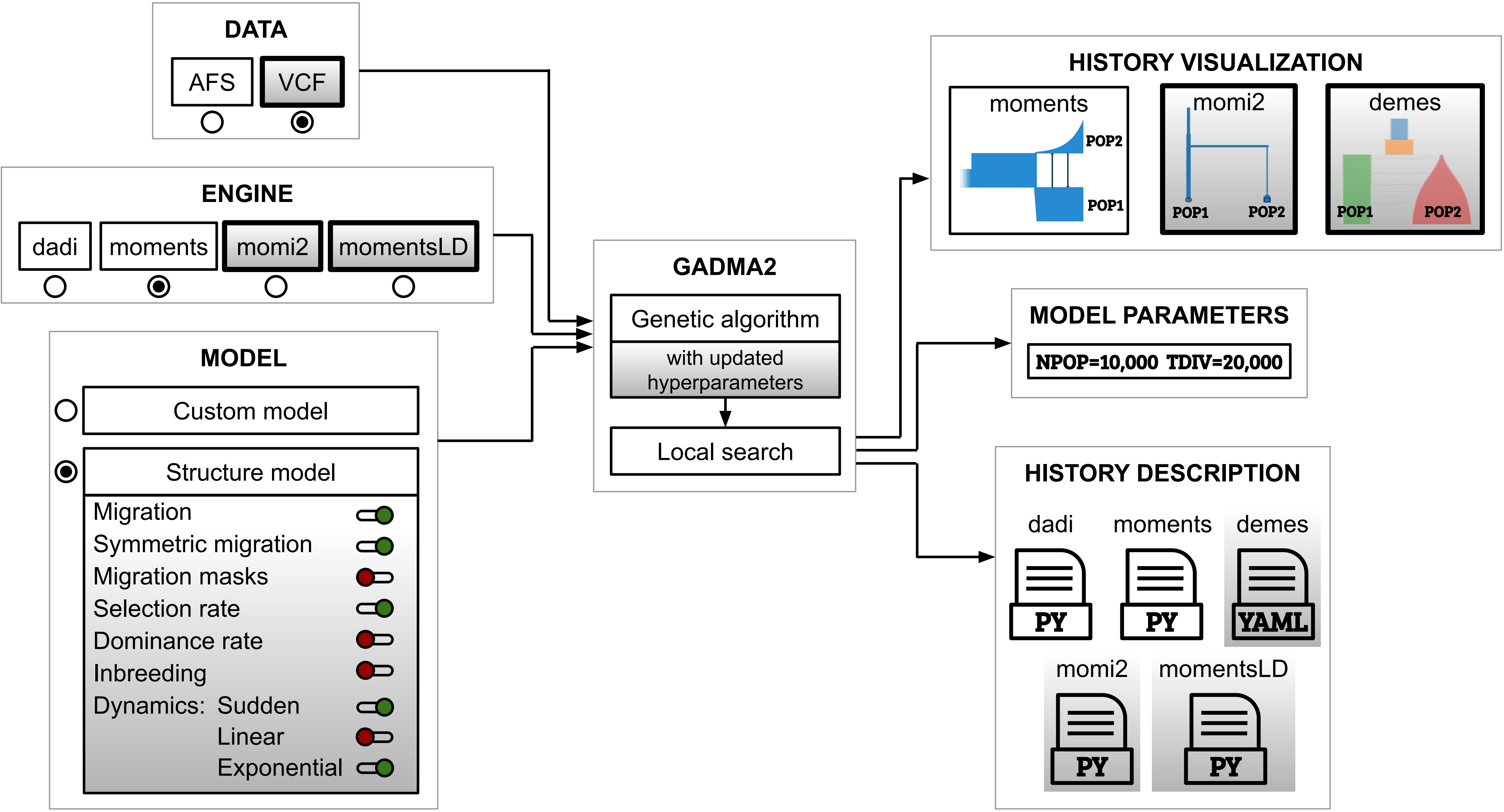
Parameter	True value	∂a∂i	*moments*	*momentsLD*
N_{anc}	17,934	17,864	17,945	19,775
N_{Bor_split}	10,617	10,787	10,246	11,187
N_{Sum_split}	7,317	7,564	7,216	7,818
N_{Bor}	8,805	9,242	9,036	9,753
N_{Sum}	37,661	38,800	37,839	41,378
$m_{Bor-Sum}(\times 10^{-5})$	0.66	0.66	0.66	0.61
$m_{Sum-Bor}(\times 10^{-5})$	1.10	1.06	1.07	0.98
T_{split} (gen.)	20,157	20,847	19,916	21,516

N_{anc}: size of ancestral population; N_{Bor_split}: size of *Pongo pygmaeus* at split; N_{Sum_split}: size of *Pongo abelii* at split; N_{Bor}: size of *Pongo pygmaeus* after exponential decline; N_{Sum}: size of *Pongo abelii* after exponential growth; $m_{Bor-Sum}$: migration rate from *Pongo pygmaeus* population to *Pongo abelii* population; $m_{Sum-Bor}$: migration rate from *Pongo abelii* population to *Pongo pygmaeus* population; T_{split}: time of divergence in generations.

Table A6: The demographic parameters of orang-utan histories with pulse migrations inferred with *momi2* engine in GADMA2. The time interval after divergence was divided in equal parts and pulse migrations were integrated between them. The inferred parameters showed convergence to true values with increase of pulse migration number. True values are the simulated parameter values that were obtained from the original paper Locke et al. (2011).

Parameter	True value	Model 1	Model 2	Model3	Model 4
Number of pulse migrations	0 (continuous)	0	1	3	7
N_{anc}	17,934	19,331	19,220	18,461	18,038
N_{Bor_split}	10,617	6,187	8,731	8,715	10,275
N_{Sum_split}	7,317	7,719	4,165	5,412	6,570
N_{Bor}	8,805	10,663	9,631	9,640	8,926
N_{Sum}	37,661	54,184	59,929	43,123	38,623
$m_{Bor-Sum}$	0.66×10^{-5}	0	0.065	0.057	0.022
$m_{Sum-Bor}$	1.10×10^{-5}	0	0.206	0.084	0.035
T_{split} (gen.)	20,157	11,270	16,211	20,086	20,538

N_{anc}: size of ancestral population; N_{Bor_split}: size of *Pongo pygmaeus* at split; N_{Sum_split}: size of *Pongo abelii* at split; N_{Bor}: size of *Pongo pygmaeus* after exponential decline; N_{Sum}: size of *Pongo abelii* after exponential growth; $m_{Bor-Sum}$: migration rate from *Pongo pygmaeus* population to *Pongo abelii* population; $m_{Sum-Bor}$: migration rate from *Pongo abelii* population to *Pongo pygmaeus* population; T_{split}: time of divergence in generations.


DATA
AFS
VCF
ENGINE
dadi
moments
momi2
momentsLD
MODEL
Custom model
Structure model
Migration
Symmetric migration
Migration masks
Selection rate
Dominance rate
Inbreeding
Dynamics: Sudden
Linear
Exponential
GADMA2
Genetic algorithm
with updated hyperparameters
Local search
HISTORY VISUALIZATION
moments
POP2
POP1
momi2
POP1
POP2
demes
POP1
POP2
MODEL PARAMETERS
NPOP=10,000 TDIV=20,000
HISTORY DESCRIPTION
dadi
PY
moments
PY
demes
YAML
momi2
PY
momentsLD
PY