Master 2 Bioinformatique

Semestre 10

Université de Bordeaux

Project report: Statistical Analysis Pipeline for Admixture Data from a Human Population Settlement Model

Aurélien Luciani

Supervisor: Murray P. Cox

Host: Institute of Fundamental Sciences, Massey University

Université de Bordeaux supervisor: Marie Noelle Beurton-Aimar

Year 2014-2015

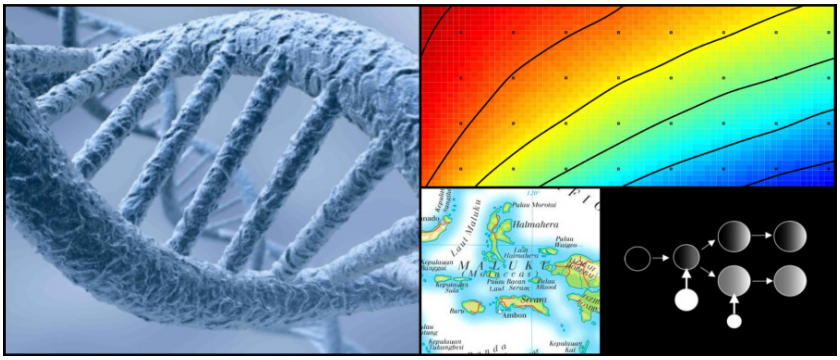
Contents

Acknowledgements

Special thanks to:

- Dr Murray Cox, as the main supervisor of this project and of this internship;
- François Vallée, as the research assistant for this project, and developer of the model;
- Dr Marie Noelle Beurton-Aimar, as the Université de Bordeaux supervisor of this internship; *and*
- all the Computational Biology Research Group, Dr Matthew "Mac" Campbell, Dr Pierre-Yves Dupont, Dr Elsa Guillot, and Jiří Moravec, for the help provided during this internship.

Computational Biology Research Group: `http://massey.genomicus.com`



This work is founded by the Rutherford Foundation Trust, part of the Royal Society of New Zealand. It is hosted at the Institute of Fundamental Sciences (IFS), Massey University, Palmerston North, New Zealand.

Introduction

Two main ancestral populations can be identified in the genomes of people living nowadays in the Islands of South-East Asia (ISEA). The first one is the Melanesian population, whose settlement in this area is believed to have taken place around 45 thousand years ago. The other arrived more recently, during a period often called the Austronesian expansion, between 5 and 4 thousand years ago, when people from what is now mainland China settled in the islands. This expansion implied great changes for the people living in these islands, including the development of rice agriculture and better navigation knowledge. The exact path these populations followed is unclear though, and it is believed they might have come through Taiwan, the Malay Peninsula, or from both places around the same time.

Nowadays, people living in this area have mixed genomic ancestry and markers can be identified as either from Asian or Melanesian ancestry. A measure of this is the **admixture** rate that will be used throughout the project, providing a simple way to define the ancestry of an individual. The markers used are based on single nucleotide polymorphisms (SNPs) located on different chromosomes and 37 markers can be used to define accurately the admixture of the Asian ancestry in every individual (Cox et al. [2010]). The choice of this set of SNPs is the result of previous studies and the resulting set is defined as highly informative. With this, the ancestry of a person can be defined based on a small quantity of markers that are highly discriminant.

Two specific patterns can be seen (Lansing et al. [2011]). One is the non-linear gradient of Asian admixture along the longitudinal axis in the different islands in the area, that could correspond more or less to how the Austronesian expansion wave propagated

from the mainland. The second is the difference of admixture when looking at specific parts of the genomes associated with male or female ancestry, implying a gender-biased expansion.

These are based on data coming from studies made on populations currently living in the ISEA. These studies were made jointly at Massey University (New Zealand), the University of Arizona (USA), the Santa Fe Institute (USA), and the Eijkman Institute (Indonesia). They provided the set of SNPs used, determined after the genotyping of 1,430 individuals from 60 populations. The data and discoveries made led to further questions, such as explaining the different admixture patterns.

The Computational Biology Research Group at Massey University (Manawatū campus, in Palmerston North) is doing a follow-up two-year project. It consists of developing a model of the Austronesian expansion throughout the ISEA that could reproduce the same two patterns observed in the real data. A part of it, corresponding to this internship, will be to assess the quality of this model and to analyse its results.

Chapter 1.

Project presentation

The scope of this six-month internship is to create a statistical framework to handle the data created by the model.

As the model is currently being implemented in the same research group, the first thing to do is to assess the quality of the model by looking at the returned data. This is done to guide the development of the model to have the best implementation possible.

Once the model implementation is stable, it can be run to provide data for the statistical analysis. The final goal is thus to compare the simulated data to the reference data, from the previous papers (Cox et al. [2010]; Lansing et al. [2011]), and to find the same patterns observed with the reference data in the simulations.

This project will need to take into account what is already done to evaluate simulation data and to adapt it to this specific model in order to handle transparently all the simulation data generated.

The first of the two observed patterns can be seen in Figure 1.1. It is a non-linear gradient in Asian admixture that declines abruptly around the eastern part of Indonesia. The second one, represented in Figure 1.2, shows the difference in admixture on distinct parts of the DNA for different islands. These latter differences would tend to appear in a contexts where marriages between Asian women and Melanesian men would be favoured.

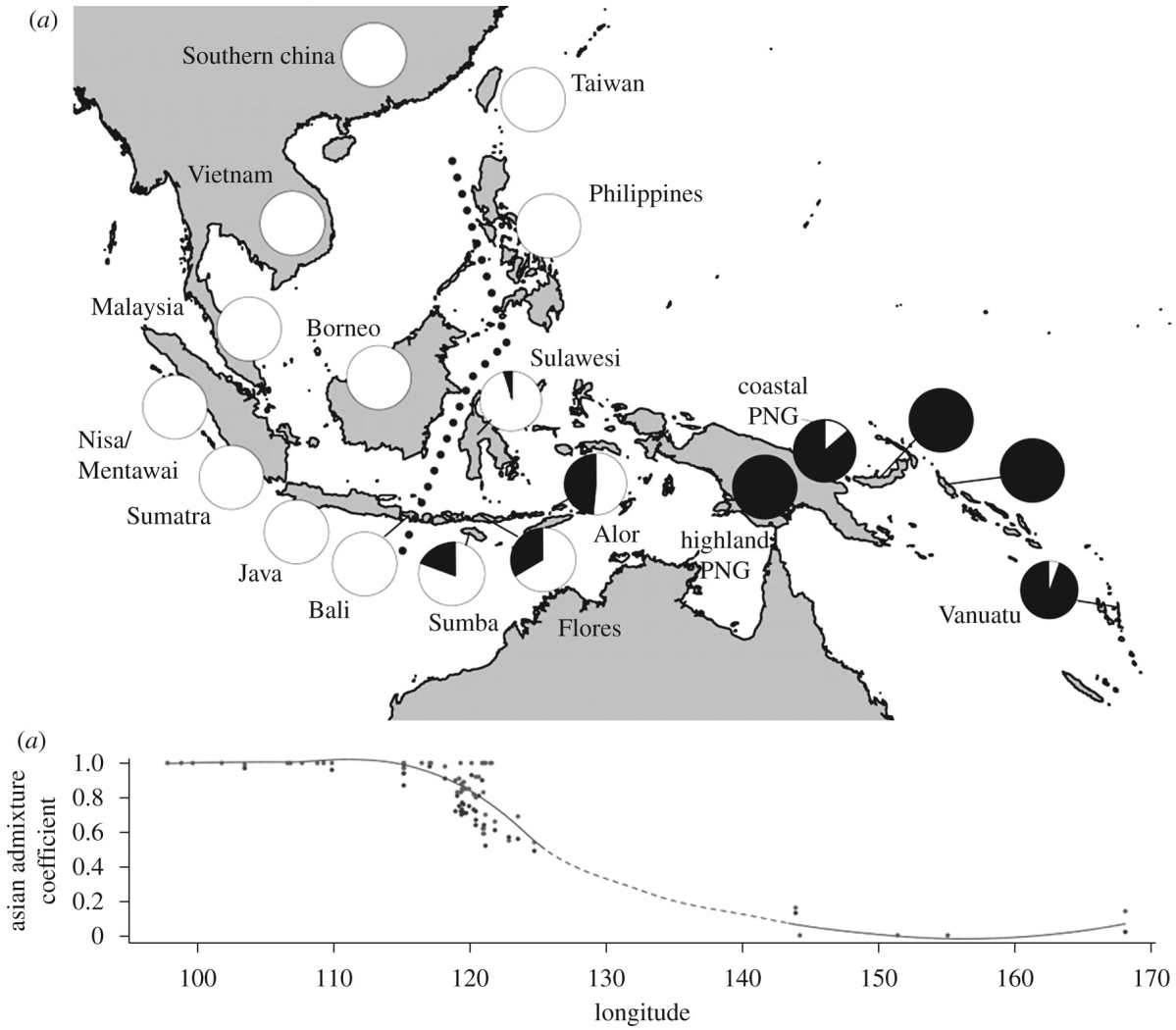


Figure 1.1.: *Local admixture rates across the Indo-Pacific region. (a) Pie charts showing mean regional admixture rates (Asian component in white; Melanesian component in black). Wallace's biogeographic line is shown as a dotted line. Regional admixture rates are shown for data reduction purposes; (b) Change in Asian admixture rates calculated from all SNPs combined (black line). Regions with no data indicated by a dashed line (exact gradient unknown). Asian admixture estimated from autosomal and X-chromosomal SNPs are indicated by black and grey points, respectively. Note the decline in Asian admixture beginning in eastern Indonesia, as well as preferential retention of X-chromosomal (grey) versus autosomal (black) diversity. Reproduced unmodified from Cox et al. [2010].*

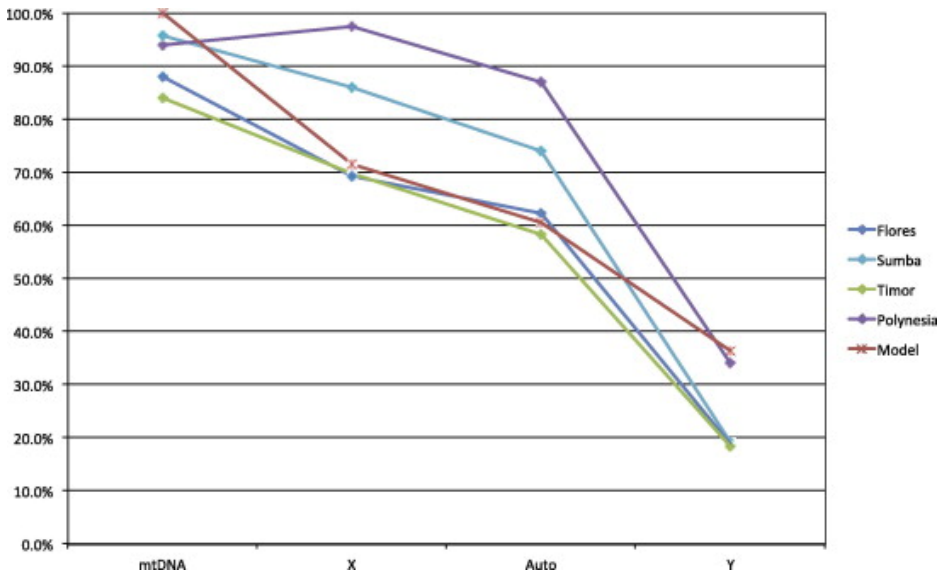


Figure 1.2.: *Fraction of Asian DNA in four genetic systems compared with model results for* $\alpha = 0.02$ *and 50 generations. Sample sizes: Flores = 453, Sumba = 639, Timor = 529. Reproduced unmodified from Lansing et al. [2011].*

The whole idea of the project is thus to reproduce the same patterns and to determine which sets of parameters lead to outputs similar to what is observed. The model is set to run from the beginning of the Austronesian expansion, around 4,500 years ago, to the present.

The simulated data from the model will be the observed admixture rates of the populations in the area covered by the model. The admixture will be measured for five parts of the DNA: the whole DNA, the mitochondrial DNA, the autosomal DNA and the X and Y-chromosomal DNAs. These admixture values will only be recorded for the last step of the simulation, since they will be the only values comparable to the real observed admixture results.

Chapter 2.

Existing context

2.1. Agent-Based Model

The model is an Agent-Based Model (ABM) that comprises a graph of nodes, each node corresponding to a **deme**, and each edge corresponding to a possible migration route between demes (see Figure 2.1). The agents of the model are people living in the graph, in every node, and these agents are able to migrate and to create families. The graph and the agents living in it respond to a set of basic rules regulated by parameters and the goal is then to observe the emerging behaviours arising from it.

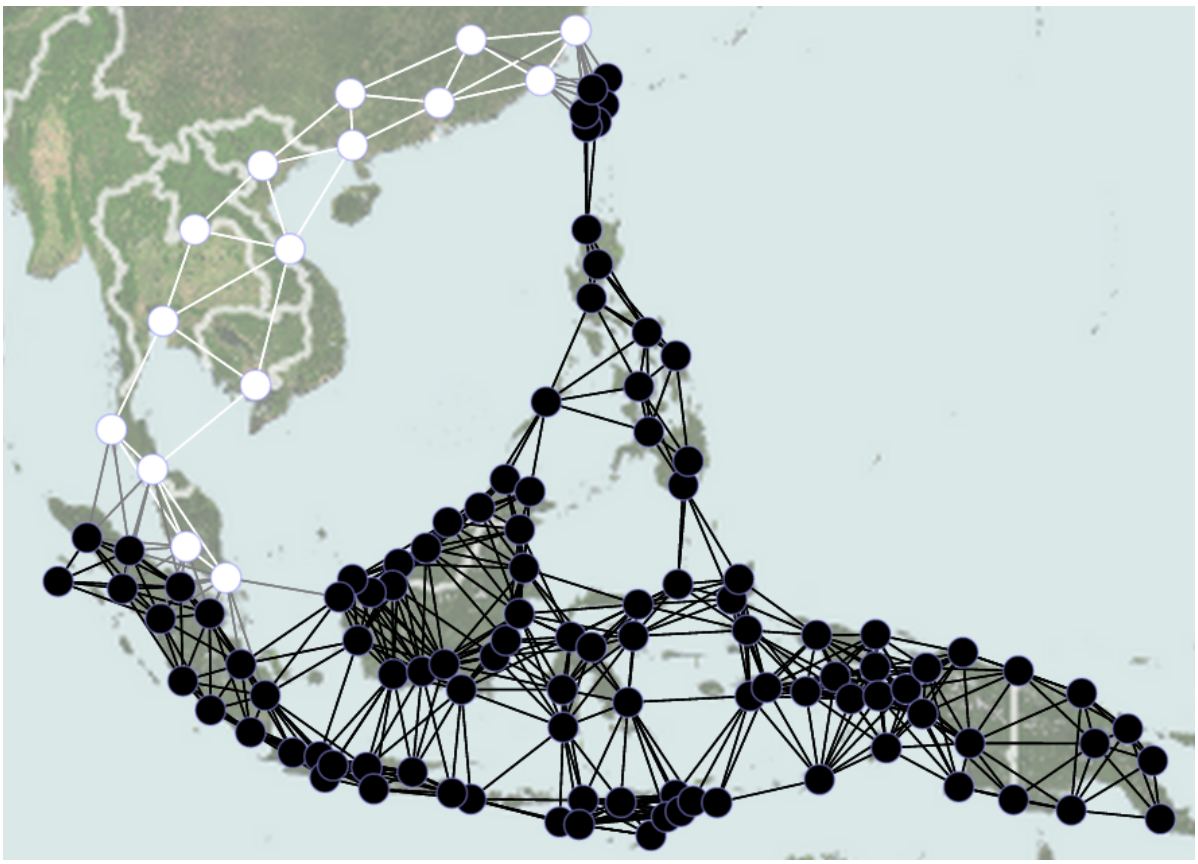


Figure 2.1.: *Nodes of a model superimposed onto the map of the modelled region. White nodes are the Asian nodes at the beginning of the simulation and black nodes are the Melanesian ones, according to one of the possible starting distributions.*

Because of the stochasticity of the model, and because the parameters used might not be viable, a simulation can end up having non-usable outputs. For example, if an island ends up being completely empty, the admixture values yielded by this specific simulation will be the value `NaN` (not a number). In such extreme cases, the model can be considered as "failed" and can either be discarded or given the worst possible score, depending on the current analysis. Arbitrarily, two rules have been set to define that a simulation has failed:

- If a deme has a population of less than 10% of the most populous deme in the network, it is considered as empty;
- If more than 25% of the demes in any of the islands are empty, the simulation is considered as failed.

The model itself has been developed in Java using the Repast Simphony framework (North et al. [2013]), a cross-platform framework made to write flexible agent-based models. The agents correspond either to a single person or to a couple that evolve in the demes. Each deme is a node in the graph and possible migration paths between demes are edges of this graph. Therefore, the implementation of the model consists of agents evolving in a graph.

2.1.1. Parameters

Numerous parameters can be set in the model. Some of them are defined initially and never change because there is no need to do so. These parameters correspond to wanted characteristics of the model in the observed context like natality or mortality rates or the age distribution. The values used come from a wide variety of previous anthropological studies.

The parameters that can actually be adjusted are listed in Table 2.1 and can be put into different groups. Most of them are continuous parameters, whose values are numbers, and a few of them are discrete. The discrete parameters are the graphs used and the starting distributions. While the former correspond to the topology of the graph used in the simulations, comprising the nodes and the edges between them, the latter are the distributions of the Melanesian and Asian populations in the graph at the initial step of a simulation.

The model can work with a large number of different parameter values, corresponding to the "values" columns in Table 2.1. Knowing the context of the model, especially using previous anthropological studies, an estimation of possible realistic values can be found, but will need to be refined later on during the analysis processes.

Parameter	Values	Estimated	Comment
Migration prob.	$\mathbb{R}_{0\leq x\leq 1}$	$\mathbb{R}_{0<x\leq 1}$	prob. to start migrating for a Melanesian agent
Migration prob. ratio	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{1\leq x\leq 4}$	corresponding ratio for an Asian agent
Fecundity	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{2.5<x<8}$	Poisson law mean for a Melanesian agent
Fecundity ratio	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{1\leq x\leq 2}$	corresponding ratio for an Asian agent
Marriage threshold	$\mathbb{R}_{-0.25\leq x\leq 0.25}$	$\mathbb{R}_{0\leq x\leq 0.25}$	affects marriages rules
Growth rate	$\mathbb{R}_{0\leq x\leq 1}$	$\mathbb{R}_{0<x<0.001}$	limiting rate of pop. growth
Number of agents	$\mathbb{Z}_{\geq 0}$	$\mathbb{Z}_{100\leq x<400}$	pop. size in each deme, initially
Graph	{...}	{...}	composition of the graph (nodes and edges)
Starting distribution	{...}	{...}	distribution of pop. in the graph

Table 2.1.: *Summary of the changeable model parameters.*

2.1.2. Other existing models

This model is just one of the types of models that could have been chosen.

This model is "forward-in-time", meaning that the simulation starts in the past and evolves chronologically towards the future. This is by opposition to coalescent models where the model evolves from a given point in time back to a previous state. These models take the problem the other way around by trying to rebuild the past, the ancestry of the individuals, based on the present data. They do it this way to try to reduce the amount of computation needed, as they therefore do not have to simulate agents whose lineage would have ended up extinct.

It is not suited for this agent-based model though because parameters vary among agents and this is not trivial to do with coalescence. The fact that an agent has either a Melanesian or Asian ancestry leads to different behaviour and this is more straightforward to do in an "forward-in-time" model.

Also, other models can have their time unit defined as a generation (roughly equivalent

to 20 years). Here, the unit of time has been defined as one year to have more precise time steps in the model and to be closer to reality due to the higher time resolution.

2.2. Statistical analysis framework

Because of the complexity of the model (number of agents, stochasticity, multiple parameters), it will need to be run thousands of times and with a lot of different changing values for the parameters. This will lead to the use of a robust statistical analysis frameworks to extract meaningful information from the high quantity of generated data.

2.2.1. Comparisons

To compare two different scenarios, comparison functions had to be defined. These functions will be able to provide a value of similarity or dissimilarity when provided with observable values that characterise the scenarios. In this case, the comparison is done between the real observed data and the output of one simulation. Every simulation output has to be treated in the same way than the observed data. Doing so allows them to be compared. The observed data do not include values for specific islands even though they are included in the simulations (see Table 2.2). This is because nodes had to be placed in the model so that the agents could migrate, even though the empirical studies did not get samples from the corresponding areas. The admixture values for mitochondrion and Y-chromosome are also not available for every island. Those values in the model outputs can thus be dropped as they cannot be compared.

Island	Simulated	Observed	Island	Simulated	Observed
Alor	✓	✓	Philippines	✓	✓
Aru	✓		Seram	✓	
Bali	✓	✓	Sulawesi	✓	✓
Borneo	✓	✓	Sumatra	✓	✓
Bougainville		✓	Sumba	✓	✓
China	✓	✓	Taiwan	✓	✓
Flores	✓	✓	Tanimbar	✓	
Halmahera	✓	✓	Thailand	✓	
Java	✓	✓	Timor	✓	✓
Laos	✓		Vanuatu		✓
Malaysia	✓	✓	Vietnam	✓	✓
New Guinea	✓	✓			

Table 2.2.: *List of islands (or areas) in the model and in the reference data.*

The first, and most straightforward way to compare the two sets of data was to do a pairwise distance between real and simulated values. But doing so would mean dropping a lot of information, mainly regarding the relative geographical position of every island in the region. Another measure was needed that would keep this information and take it into account while comparing the data. Also, the idea arose to compare the whole matrix of differences in admixture among islands instead of the values island by island.

Sixteen comparison functions were tested with randomly generated data. Seven were distances, including weighted versions of the comparisons. Nine were correlation measurements, and different correlation coefficients were tested.

These randomly generated data had the same structure as the simulated data that would be later compared but the actual values, while still being admixture values (a

proportion value between zero and one), were completely random. When comparing the results of the comparison functions with the same random data, functions that would yield correlated results would convey the same information. It would thus make no sense to choose two functions whose results are highly correlated because it would mean adding dimensions to the comparison values while not adding any meaningful information.

Two measures have been selected that can hold different information about a comparison: the mean square distance (MSD) and the partial Mantel test. They do not have a good correlation when applying them to random admixture values, thus meaning that they do not carry the same information. They are complementary, and using them both gives a better overview of the comparison.

Mean square distance

The mean square distance is the mean value of all of the distances between observed and simulated values of admixture for every n island.

$$MSD = \frac{\sum_{i=1}^{n} (AdReal_i - AdSim_i)^2}{n} \tag{2.1}$$

where $AdReal$ is the array of admixture data observed in the real values and $AdSim$ the corresponding values in the simulated model.

It gives a distance value, with zero meaning that the two observed values are absolutely identical. In this specific context, taking into account that the compared values are admixture values, ranging from zero to one, and because of the values observed in the real data, the highest possible MSD value will be around 0.8 (see reference case lines in the upper part of Figure A.3).

Partial Mantel test

The Mantel tests have been developed to compare two matrices with the same information. The partial Mantel test also uses a third matrix, holding constant geographical distances in order to be able to weight the values according to the actual geographical

distances between the points (Smouse et al. [1986]).

In this case, the first two matrices contain the values of distances in admixture between every island in the graph and a matrix M is calculated as

$$M = \begin{bmatrix} d\left(Ad_0, Ad_0\right) & \cdots & d\left(Ad_0, Ad_n\right) \\ \vdots & \ddots & \vdots \\ d\left(Ad_n, Ad_0\right) & \cdots & d\left(Ad_n, Ad_n\right) \end{bmatrix} \tag{2.2}$$

where d is the function returning the distance between the two arguments and Ad_n is the admixture of the n^{th} island in the graph. With the corresponding matrices for the simulated data, the real data and also the geographical distances, the partial Mantel test can be done as such

$$correlation = partial.Mantel\left(M_{Simulated}, M_{Real}, M_{geographical}\right) \tag{2.3}$$

The partial Mantel test returns a correlation value between -1 and 1 with a value of 0 meaning that the two matrices are not correlated at all and of 1 if they are completely correlated.

2.2.2. Approximate Bayesian Computation

A fairly recent statistical analysis framework has been used more and more in the context of population simulations. This framework is called Approximate Bayesian Computation (ABC). Its use is recommended when using high dimensional data that makes it difficult to use standard statistical frameworks. ABC still relies highly on standard statistical tools but instead of giving one best result, it returns a distribution of best possible parameters. It answers the question: given a set of parameter distributions (called "**priors**"), what are the distributions (subsets of the priors), that give best results? This set of output distributions is then called "**posteriors**".

Its use has been increasing recently and some examples of recent papers using ABC

in similar contexts are Guillot et al. [2015] or Kehdy et al. [2015].

The whole framework encompasses the steps from the choice of the parameter sets to the inference of the posterior distributions. Figure 2.2 summarises the successive steps of this process with hypothetical parameters α, β and γ and summary statistics $SS1$ and $SS2$. The parameters will be used by the model to generate results. These results are, either directly or after transformation, the summary statistics that the ABC will use. By comparing those to the references, the framework is able to keep or discard a simulation during what is called the rejection step. the parameters whose simulations got accepted can be used and visualised in distributions.

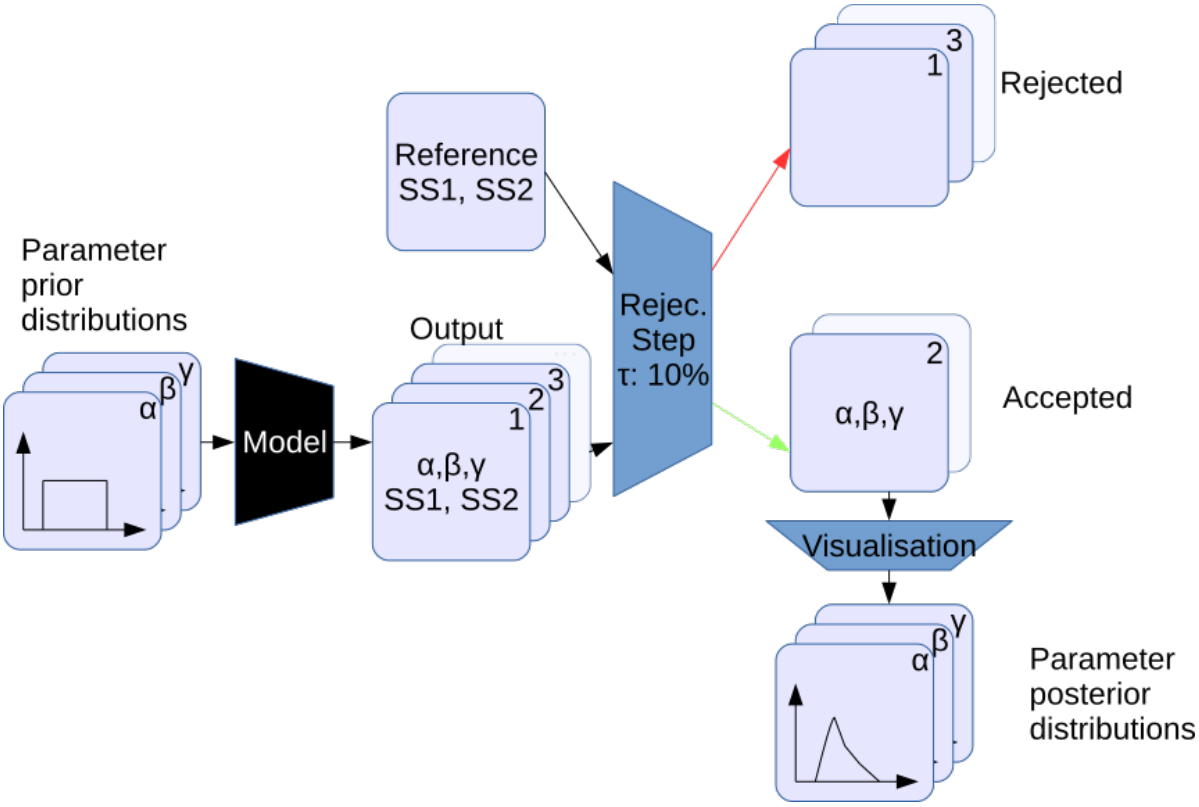


Figure 2.2.: *Visual representation of the different steps of an ABC inference framework.*

The priors are defined as a set of distributions, one for each observed parameter. The distributions can be of any type but it is necessary to know them when doing the inference of the posterior distributions since, depending on the type chosen, they can induce a bias that will need to be corrected. In general, for ABCs, uniform, Poisson or

normal distributions are used. Here, the distributions chosen are all uniform and the only bias can be if the correct parameters lay outside of the boundaries of the distributions (thus the importance of doing other previous statistical analyses). A visualisation of example simulation parameter sets that can be generated by the first step of an ABC framework can be seen in Figure 2.3.

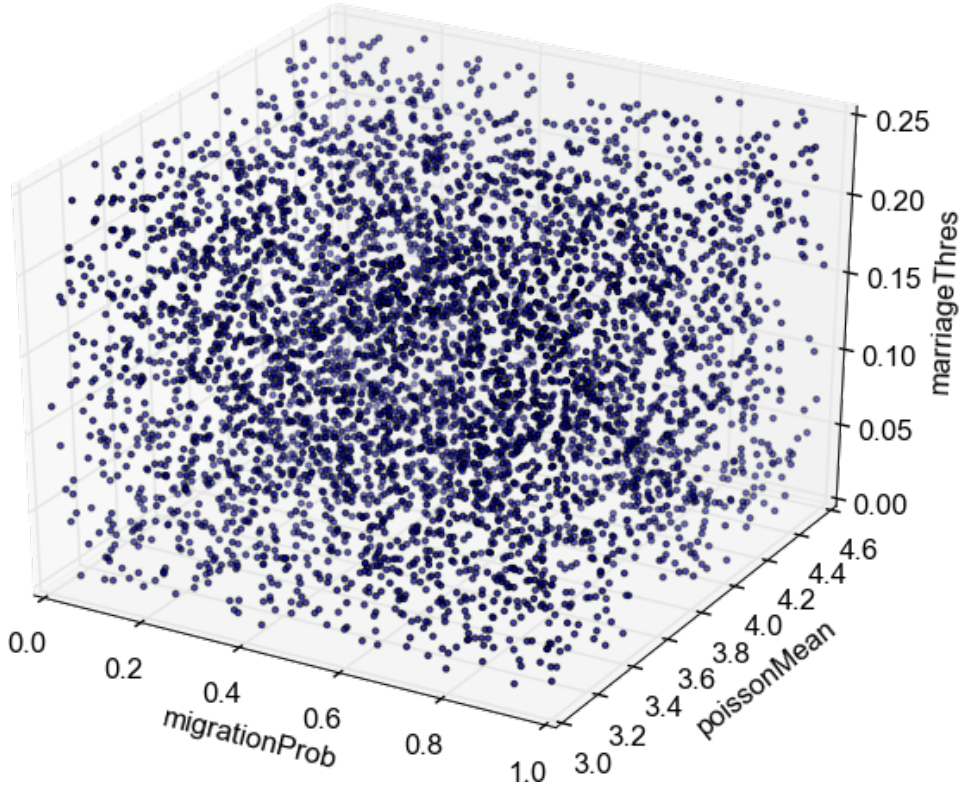


Figure 2.3.: *Visual representation of 5,000 sets of three parameters, migrationProb, poissonMean and marriageThres drawn from independent uniform distributions. As a result, the points are evenly spread inside a cuboid. Each point represents one set of parameters that will be used for one single simulation.*

Now that the required parts are defined, their implementation is needed, taking into account the specifics of the project. A pipeline is required to run the simulations and to process and analyse all the simulation data coming from the model.

Chapter 3.

Implementation

The different parts of the statistical analysis pipeline implementation are detailed in the next sections. A global overview of the pipeline is available in Figure 3.1 and presents the different modules of the pipeline, the language in which they were written, as well as the formats of the intermediary storage steps. The modular design is necessary because the analysis pipeline requires high flexibility. The arrows in Figure 3.1 only represent a few of the possible ways the pipeline can be used, and depending on what is necessary, some intermediary steps can be bypassed. Also, with the same idea of flexibility and ease of use, every script can be called independently and provides command line parameters defined in a standard way (POSIX[1] guidelines) and a corresponding help option to detail them.

The model here has been simplified to a single step in the pipeline even though it is fairly complex. This is because the implementation of the model, while it is really important for the project, does not enter in the scope of the internship. Its inner functioning and design is nevertheless required for the implementation of the rest of the pipeline.

[1]Portable Operating System Interface

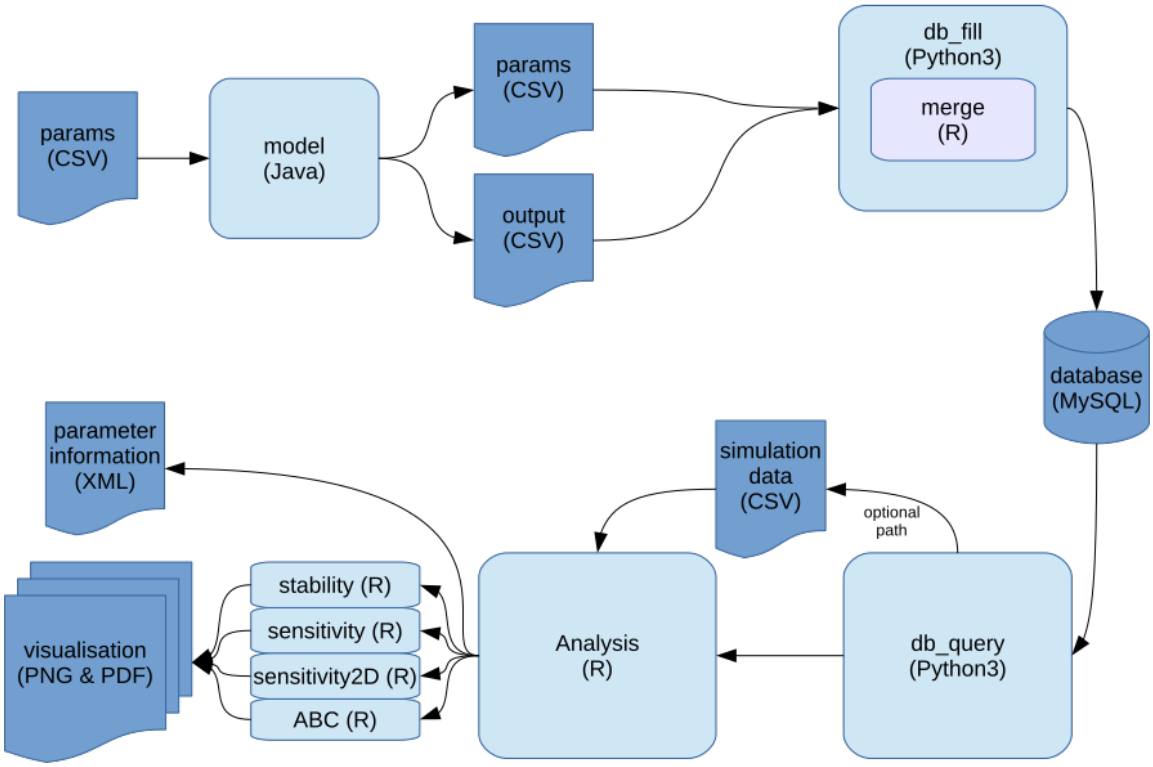


Figure 3.1.: *Visual representation of the different parts of the treatment and analysis pipeline. Dark colours represents data and the way it is stored, and light colours represent programs and scripts and the language in which they have been written.*

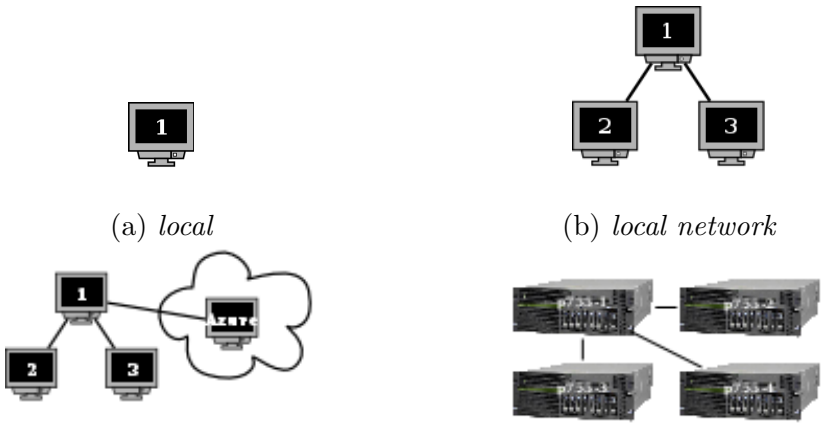
3.1. Run management

The fact that many simulations are required to infer meaningful information from the model implies that they cannot be run sequentially on a single desktop computer in a reasonable amount of time. Luckily, every simulation being independent from the others, there are multiple ways to generate more results in less time. Firstly, if the computer used has multiple processors, it can run multiple simulation concurrently. Also, a cluster of computers can be used and the simulations needed can be dispatched among the nodes of the cluster so that they each run the simulations they were assigned. When all of the nodes have ended their runs, their outputs can be aggregated and/or stored.

Different levels have been used. First, running the simulations locally (Figure 3.2a), on a single computer, then using the three computers in the office as a cluster of compute

nodes (Figure 3.2b), for more heavy batches. Also, since Massey University just made an agreement with Microsoft Azure, which provides computing "in the cloud", simulations have been run transparently by adding virtual machines on the Microsoft Azure system to the cluster of computers in the office (Figure 3.2c). Finally, when the computation requirements were too high, for really huge batches, NeSI[2]'s High Performance Computing (HPC) facilities have been used (Figure 3.2d). They provide servers specially designed for scientific computation and can be tasked with hundreds of parallel jobs at a time and multiple batches can be queued so they will be run as soon as computational power is available. Each node is an IBM Power755 machine.

The most powerful level used for this study is obviously the HPC but a trade-off of using this system is that, since it is shared by multiple users and is managed by a third-party, it requires specific settings and it cannot be used exactly as can a custom cluster of computers. The batches have to be submitted through a **job scheduler**, in this case called LoadLeveler. Because of this, slight changes have been made to the way the model and the Repast framework are launched. Simulations run on this HPC use several nodes, with each of these nodes having multiple threads available for the simulations. This configuration leads to the parallel simulation of more than a hundred scenarios at the same time. In this case, 4 nodes were used, each of those having 32 threads.



(a) *local* (b) *local network*

(c) *local network and Azure* (d) *NeSI*

Figure 3.2.: *Different set-ups to run the model.*

[2]New Zealand eScience Infrastructure: `https://www.nesi.org.nz/`

3.2. Data processing, storage and querying

Since running a lot of simulations, with a lot of different parameters, can generate a lot of output files whose original parameters can be hard to track, a way to keep them organised was needed. Doing so, it would also be possible to use results from different batches and to analyse them together, thus avoiding having to redo simulations for parameter values already tested and reducing the computational resources used.

The organised way to store this is naturally in a database. The choice has been made to use a relational database. The structure and indices have been designed so that the parameters can be efficiently queried. The tables and relations between them can be seen in Figure 3.3. It has first been developed and tested locally and then, once it was working properly, it was deployed on a MySQL database server provided by Massey University for research purposes.

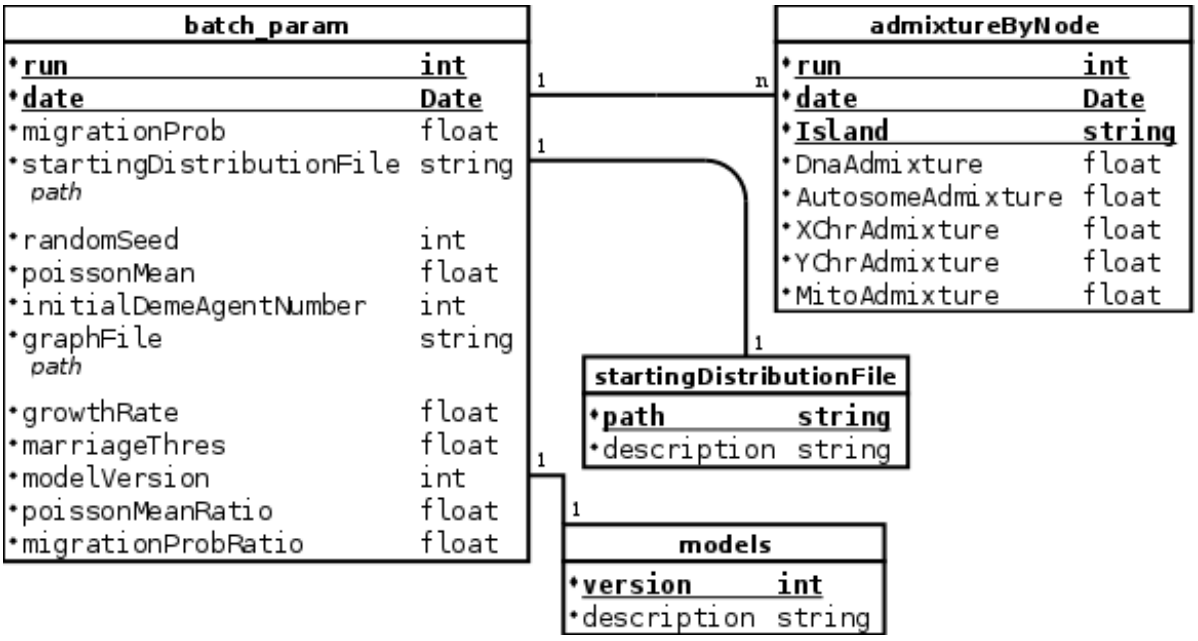


Figure 3.3.: *Structure of the database storing the results of the simulations and the corresponding parameters.*

The use of the database can be done through phpMyAdmin but two scripts have been developed to interact with it directly (`db_fill` and `db_query` in Figure 3.1). The first one reads the two files output by the model (the admixture data and the corresponding

parameter values), merges them and adds the resulting data to the database. The other one queries the database according to conditions provided by the analyst and returns already merged parameter and admixture values. The scripts have been written in Python3, using the PyMySQL module to access the MySQL database. Internally, the first script to add data to the database uses another script, in R, that processes the data before adding them. This is the `merge` script in Figure 3.1. This way one process, the Python script, handles the connection with the database while another process, the R script, does the actual data processing at the same time, in parallel.

3.3. Statistical analyses

Grid search

In order to reduce the parameter space, a grid search has been performed. It consists of going through the possible parameter sets by setting parameter values at regular steps throughout the space and running n multiple simulations for every point in the space to obtain statistically meaningful results for every point. One parameter grid search can be seen in Figure 3.4.

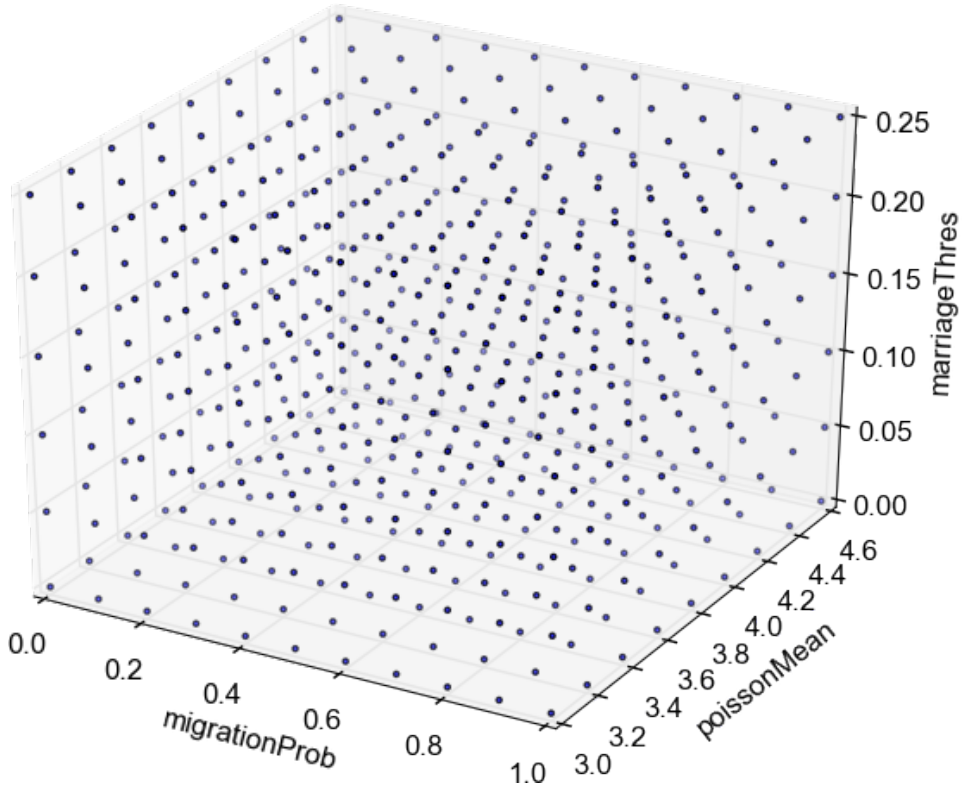


Figure 3.4.: *Visual representation of a grid search on three parameters, migrationProb, poissonMean and marriageThres. Each point represents a set of those parameters, that will be used for n different simulations.*

Depending on the number of parameters and the step sizes, this has an important computational cost. This has been considered a necessary step though, in order to reduce the parameter space for the next analysis, the ABC analysis, that would be even more computationally consuming if run over a huge parameter space. The resulting parameter space that will be used after the grid search analysis is defined manually after selecting interesting and meaningful values that can either be a smaller space or a specific value, thus making a parameter a non-changing one. Details of this will be presented in Section 4.1.

The corresponding visualisations are generated by the `stability`, `sensitivity` and `sensitivity2D` R scripts in Figure 3.1. These scripts only generate the graphical visualisations, in order to separate the logic in the pipeline between statistical analysis and data visualisation. The correct script is chosen automatically by the `analysis` script

without the need for the analyst to specify it. The choice is done dynamically during the processing of the data by this script.

ABC framework

Once an interesting subspace of parameters has been defined, it can be used to feed the ABC framework used later. Whereas a standard Bayesian analysis can be used to infer a single best value for every parameter, here the use of ABC gives a posterior distribution of parameters corresponding to the best values. The shape of the distributions can thus be appreciated and this offer better understanding of the output values of highly stochastic models run a large number of times. This technique, although quite recent, has been described in numerous studies (Sunnåker et al. [2013]; Csilléry et al. [2010]).

In this project, the parameters used are the ones defined in Table 2.1 even though the ones set after the grid search analyses will be ignored by the ABC as they are defined once and will not change. They are simply not defined by a distribution as they are constant, and running them through the ABC is useless. The summary statistics will be the four results of the comparison functions defined in section 2.2.1: the MSD and the partial correlation for both the X-chromosome and the autosomes.

At first, the R package "abc" Csillery et al. [2012] was used to understand how it worked and to be sure that the framework was used correctly. But in the end, to simplify the pipeline and to cope with the specifics of the project, custom code inspired by the R package was written. The main difference is that, instead of using the results of the comparison functions as summary statistics that would be fed into the package, these results are directly used to accept or reject a simulation. Also, changes have been done so that the ABC analysis fits better into the project and provides more useful information for this use case. For example, since the change to custom code there is no need to reshape the R data to run the rejection step, and the threshold for this step is also easier to change without rerunning the whole analysis.

The first step of the ABC consists of generating the priors. This is done via a custom Python3 script which uses a parameter file (YAML format) into which the distributions

are defined. The script generates n parameter sets that will be used as is by the model. The randomness of the generated values among the distributions is very important for the posterior inference so this aspect has been taken into account, otherwise the results could become biased.

After the model has generated results for every parameter set, the ABC framework needs to choose which simulations are closest to the reference data. In this case a rejection step with a tolerance has been implemented. For a tolerance $\tau = 10\%$ for example, the ABC will keep the 10% of the simulations whose results are closest to the reference. It is important to have a low tolerance taking into account that most of the parameter space (the borders of the space) is expected to give "bad" results. The tolerance can be adjusted for every analysis and no perfect tolerance exists; this instead has to be defined by the analyst using the ABC.

Finally, the parameter values for the selected simulations are retrieved and the corresponding distributions form the posteriors. These inferred distributions are the actual result of the ABC and can be used to deduce which parameter values, and which ranges, are more likely to yield results close to the observed reality.

While the analyses for the ABC are done in the `analysis` script in Figure 3.1, the specific visualisations are generated in the `ABC` script. This is also to separate the logic of the scripts by doing the analysis in a common script and the visualisations in an other.

3.4. Visualisations

As said in Weissgerber et al. [2015], it is easy to represent data but hard to represent it so that the person visualising it does not need to dig into the huge quantity of data again to understand what is happening. It is important to represent the data well so that it is useful to the analyst. A trade-off has been found between showing as much data as possible and doing visualisations easy to understand and showing what it is expected from a specific type of visualisation.

For this, an important part was to try to avoid simplistic visualisations when they

would not show enough information. For example, a simple mean or median value is not enough when there are alternative ways to represent these data. Whenever possible, notched box-plots were used to show the distributions and shapes of the data. The centre of the box-plot represents the median, the interquartile range is represented by the box and outside of the whiskers are possible outliers from the data.

In some cases, too many box-plots would overwhelm the analyst so, as an alternative, mean values are used but with additional error bars corresponding to the standard deviation in order to quickly evaluate the significance of the differences between the displayed values.

To let the viewer easily grasp diverse data, even though the values are different and work differently, they share a single colour theme. For example, low distance values and high correlation values are both displayed in green colours, as a way to signal "good" values. Inversely, high distances and low or opposite correlations are displayed in red shades as a way to show that they represent "bad" values in the specific context of this project.

Figure 3.5 shows a part of an example visualisation showing with good and bad correlation values for the autosomal and X-chromosomal admixtures. Low correlations appear in specifics areas of the parameter spaces where both the migration rates (*migrationProb*) and the fecundity (*poissonMean*) are either low or high. A strip of high correlations is visible and reveals a trade-off between these two parameters.

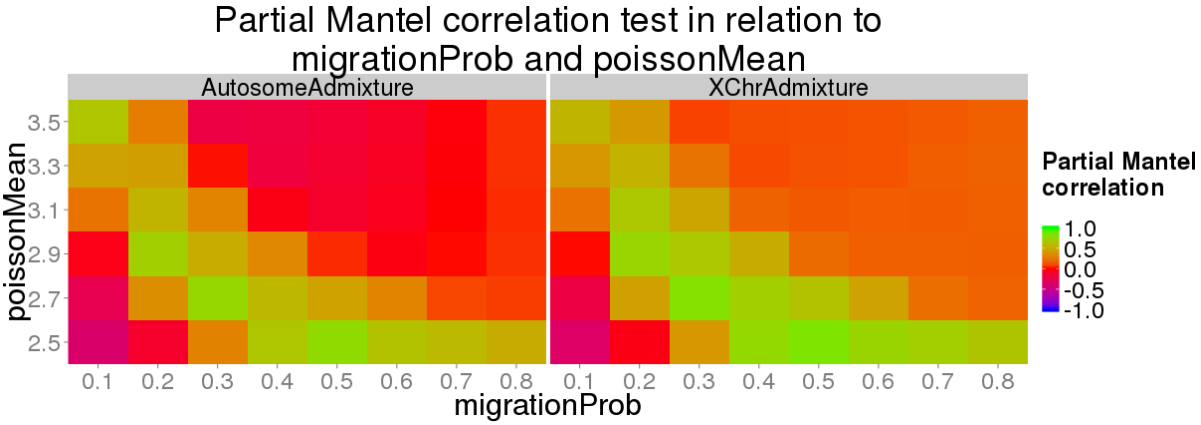


Figure 3.5.: *View of a part of an output graph, highlighting the different possible colours in the visualisation. The complete graph is in the appendices, Figure A.4.*

Examples are available in the appendix (see section A) and are made in R using the ggplot2 library (Wickham [2009]) to get nice looking graphs while having simpler and thus more maintainable code. They will be detailed while presenting the results in chapter 4.

Also, a common theme has been defined for all the visualisations to have consistent colours, fonts, and sizes among the different generated images. Finally, the images are saved in a raster format (PNG) but also in a vector format (PDF). Because of this vectorial output, changes can be applied to the images for publication, for example if a specific aspect ratio is required or if changes need to be made to the positioning of legends or scales. More importantly, this is the only way to have a lossless format that can be resized without quality loss.

Other visualisations

The visualisations were made and adapted as the project evolved so that they could be used to help make decisions and choices. They can still change and be adapted to new needs and that is supposed to be made easier through the use of adapted R libraries like ggplot2 that add a layer of abstraction above standard R code. The code responsible for the visualisations is also restricted to specific parts of the pipeline, making it highly modular and easy to change or to replace.

Until now, the graphs generated were limited to two-dimensional graphs, mainly for easier understanding and integration in papers, but one way to add more information can be through the use of three-dimensional graphs or animated graphs so that a higher number of model parameters can be visualised at the same time. It is important though that those new visualisations keep a certain level of readability that can easily be lost with higher dimensional graphs.

3.5. Optimisations

Computational cost

Some key steps in the analysis framework need to be efficient enough so that time is not lost waiting for results to be treated or for graphs to be plotted and that the computation can be done without needing a supercomputer. This project has seen a few important refactoring steps to be able to cope with the quantity of data to be treated. The most important one has been to start treating the simulation results as a stream of values instead of loading the whole dataset in memory. This was the only way to make the whole analysis process really scalable and to be able to handle a very high quantity of data.

This changed the memory complexity from linear to constant and it actually improved the time complexity from sub-quadratic to linear for the treatment part and from sub-quadratic to linearithmic ($n \log n$) for the analysis part. The improvement in time has been made possible by assuming that the results of one simulation are always together in the result stream, that way saving the cost of searching results in a big block of memory when they are actually next to each other. Actual execution times have been recorded and can be seen in appendix B in graphs B.1 for the treatment step and B.3 for the analysis step. The corresponding maximum memory use values are shown in the graphs B.2 and B.4.

There is effectively still a great amount of room to improve memory usage, especially seeing that the base memory usage is between 60 and 200 MB, depending on the script run. The base memory usage includes loading libraries, functions, global variables and also set-up and clean-up code.

The stream approach also allows the different steps of an analysis to be run simultaneously, by piping each step into the next one, effectively making the whole process runnable in parallel. This is useful only if the computer used has at least n cores if n processes need to be run in parallel. In this case, the time of the whole process is the time needed to run the longest step.

Importance of randomness

One important aspect that has been discovered while working with the Repast framework is the way it handles randomness. For a stochastic simulation, randomness is key, since starting two simulations with the same random seed and the same parameters would lead to the same succession of events in the simulation and thus to the same outcome. These two simulations would actually be identical.

When looking at the results from a statistical point of view, two identical simulations cannot be used as two distinct values. This would simply make no sense and lead to biased interpretations of the values.

The problem with Repast is that it uses the current time to generate a random seed. This can be acceptable when the program is run on one thread on a single computer, since the time at which one simulation starts will always be different from one simulation to another. It might lead to problems if the simulations are run in parallel using multiple threads on the same machine and/or using different machines and they happen to start at the same time. The risk of both the random seed and the parameter set used for the simulations colliding, while not probable, is still possible and thus not acceptable. Actually, even though the random seed is supposed to be a 32 bit signed integer, meaning that there are more that 4 billion possibilities, collisions happened more than once during this project.

The first way to handle this has been to alert the analyst when this happened so that they could remove the specific simulations if needed. Secondly, a way has been found to generate the random seeds for the ABC analysis beforehand using UNIX's random source, `/dev/urandom/`, that can generate pseudo-random values which can be used for cryptographic purposes, meaning that it is good enough in this case to avoid collisions.

At this point, all the pipeline is implemented. It has seen multiple refactoring processes to finally have the modular structure and to be able to handle data as a stream. Even though it will continue to evolve and adapt to the changes in the model and to the requirements of the statistical analyses, different interpretations can start to be given.

Chapter 4.

Statistical analysis results

4.1. Grid search analysis

Stability analysis

For the outputs from simulations with grid searches, multiple types of analysis were used, depending on the needs. The first one was to look at a specific point of the grid and assess the stability of the output generated with this parameter set.

The stability analysis outputs different views of the data for this point in the grid. The first ones are 4 graphs, one for each type of observed DNA, for which box-plots for the admixture values of each island are displayed (partial view in Fig. 4.1, complete view in Fig. A.1). The size and spread of the box-plots represent the variability of the data among the different simulations with the same input parameters.

With "correct" parameters, the admixture values are supposed to be less stable in the contact zones between the different populations at the end of the simulation than in the extremities of the graph, where the source populations live. In the simulations used to create Figure 4.1, the Philippines were a contact zone between the two populations after the 4,500 years simulated. The corresponding box-plots (the fifth from the left) are indeed more spread than most of the others.

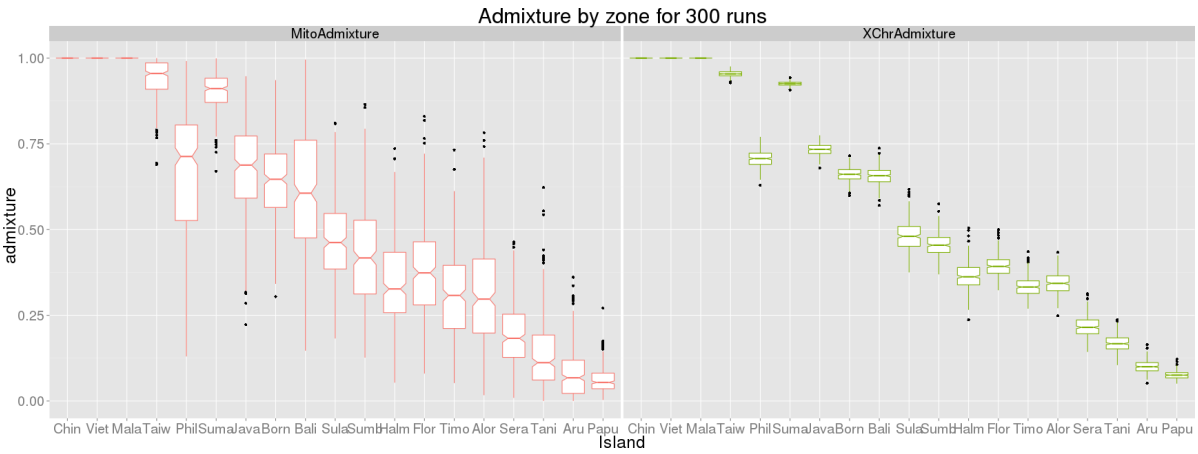


Figure 4.1.: *View of a part of an output graph, showing box-plots for mitochondrial and X-chromosomal admixture values by island. The complete figure is in the appendices, Figure A.1.*

Less stability is also expected when looking at admixture values comprising fewer markers. That is the case with mitochondrial and Y-chromosomal admixture where the corresponding DNA only have one marker each, whereas X-chromosome and autosome have 25 makers each, leading to more stable admixtures.

And finally, lower stability is expected from islands that include fewer demes, and thus lower population sizes, like Alor, Aru or Bali that are composed of only one single deme each when New Guinea (Papu in the figures) has twenty.

If the parameter value set induces abnormal behaviour, it can mean that it is not a realistic set of values or, since the goal is also to assess the model itself, it can also mean that there is a problem somewhere in the model's logic.

Another view that is output by the stability analysis is the graph of admixture values as a function of the types of DNA, and for every island (Fig. 4.2). This graph aims to resemble Figure 1.2 (Lansing et al. [2011]) as a way to compare the real data in this paper to the data generated in the simulations.

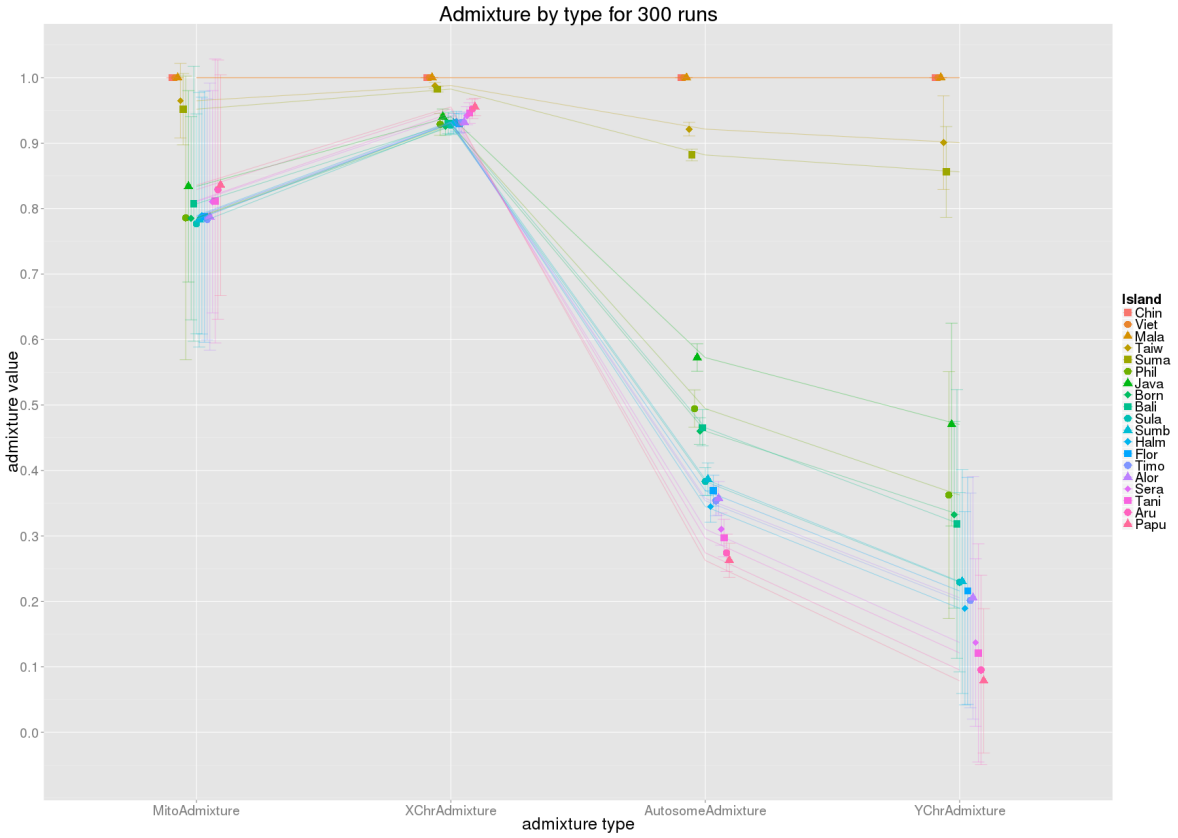


Figure 4.2.: *Admixture values by type of DNA, for every island.*

Sensitivity analysis

When looking at lines or planes on the grid, the sensitivity of the output to changes in the corresponding parameters can be analysed.

At first, this analysis was done by looking at a single changing parameter while the others were fixed at one value. This was called a one-dimensional sweep and corresponds to parameter values aligned on a one-dimensional line in the grid. Then this could be done on a two-dimensional sweep, corresponding to a plane.

Even though it was technically possible, no higher dimension was analysed because of the difficulty of further visualising the results. Indeed, the visualisation choices would have needed three or higher-dimensional visualisations and those are less straightforward to represent and to analyse. There is absolutely no limitation to the dimensionality when running the simulations, and batches were actually regularly run with more than two

dimensions. Only the visualisation has limitations with high numbers of dimensions, which is why a subset of the simulations were used when performing the visualisation.

The first thing checked is the count of simulations for every point in the grid. This is done to be sure that, during the further steps, similar number of simulations will be compared. Otherwise, and especially when there are low numbers of simulations, results can be biased and the standard deviations can be non-comparable. An example of this visualisation can be seen in Figure A.2.

Then, comparisons to the real data are performed, for both the autosomal and X-chromosomal DNA data (see section 2.2.1) and the resulting values are displayed as seen in Figure A.3 and A.4 so they can be analysed by the analyst.

Because of this analysis, specific values of migration have been discarded. Really low migration ($migrationProb < 0.1$) or nearly systematic migration ($migrationProb > 0.8$) are conditions in which it gave clearly bad results. The parameter space for the migration probability has thus been reduced to $[0.1, 0.8]$ when doing further analyses.

Also, really low fecundity rates gave too many failed simulations. Because $poissonMean$ values below 3.5 were thus considered unrealistic, the space was reduced to only include values higher than 3.5.

4.2. ABC framework

The ABC will output multiple graphs, the most interesting being the graph seen in 4.3. It shows both the priors and the posteriors of all the changing parameters in the simulations.

The red density plot is only there to remind the kind of prior distribution used and to be sure the right one was used. Indeed, if the distribution shown does not correspond to the one wanted, it can mean that there was a problem somewhere in the pipeline. If the distribution does not exactly look like the one defined, then it means that the number of simulations is too low. The implementation allows the analyst to generate an other batch of simulations that they will be able to combine to the previous one in order to

have more precise results. Also, it shows the range of values used for the ABC since it is essential, if the this range was previously reduced, that the posterior distributions remain clearly inside these limits.

The posteriors are displayed in green and correspond to the parameter distributions of the accepted simulations in the rejection step of the ABC analysis. These posterior distributions are the distributions that can be used by the analyst to make conclusions about the functioning of the modelled processes.

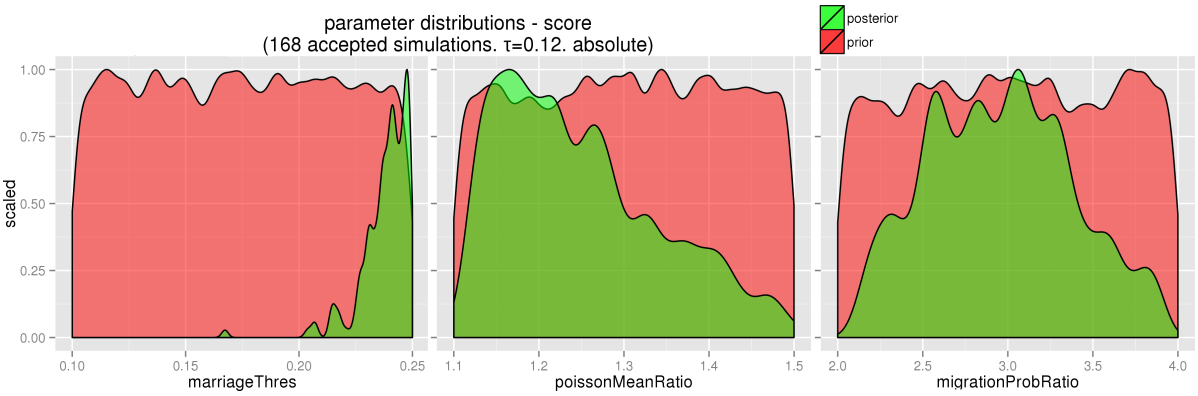


Figure 4.3.: *Prior and posterior distributions for an ABC analysis.*

Here, in Figure 4.3, by looking at the fecundity and migration ratios (`poissonMeanRatio` and `migrationProbRatio`), it can be deduced that, the Asian population had an advantage in fecundity and in migration (accepted ratios > 1) over the Melanesian populations. This is most likely due to the knowledge of rice agriculture and to better navigation techniques.

Also, the `marriageThres` posterior distribution is peaking close to the space upper limit, meaning that the interpopulation marriage rules were not neutral and likely favoured Asian women — Melanesian men marriages.

The implementation of the ABC framework being the last part of the internship, these interpretations might be subject to changes. More useful results are to be expected in the future when more ABCs will be run.

Conclusion

The model used is still undergoing changes, with the aim to make it more accurate and to take into account more factors. The problem would be to add enough complexity to keep the model relevant while still being computationally simple enough so that it can be run in an acceptable amount of time. Several aspects can be changed, ranging from the structure of agents and how they are managed in the model, to higher level tasks like the integration of resource management at different scales of the model to try to be more accurate and have more realistic behaviours.

This project will continue evolving and the work done until now will still be useful. The scripts written will still be able to handle the data output by the model, and even if the data changes, only slight changes will be necessary for them to continue working.

On an anthropological level, results have already been found. Even if research needs to be followed-up, the first findings have been able to discard some scenarios and to confirm others. For example, the marriage bias has been confirmed. Indeed, the simulations done without any bias or with an inverted bias were discarded by the ABC framework. A trade-off between fecundity and migration rate has also been shown, meaning that a good parameter set can either have high fecundity but low migration rate, high migration rate and low fecundity, or medium values for these two parameters. Any of these combinations yields simulations whose final state is similar to the real values observed in the field.

This internship has been an opportunity to work on a real research problem, to think about it and to decide what approach would be best to answer the questions raised by this problem. In order to do so, different pieces of software have been developed and will be reusable to continue the research in the same direction. The languages of the

different scripts have been chosen to use the ones adapted to a specific task; for example all the heavy data handling has been done in R. The code produced aims to easily handle the data generated by the model used for this project and to represent it in a way that conclusions can be made by looking at the visualisations of the data and not having to manually analyse the huge quantity of data generated by the numerous iterations of the model. All the data have been handled in a recent but recognised statistical context, an ABC framework, that guarantees that the interpretations made are not biased and can be found again using other implementations of the same statistical framework.

Glossary

admixture

Here, genetic admixture. Introduction of new genetic lineages into a population. Refers to the proportion of the genome coming from one ancestral population or another. In the whole context of this project, when talking about admixture, Asian admixture is implied, with a value of one for a completely Asian person, and zero for a person having absolutely no Asian SNPs, and since only two populations are considered, completely Melanesian. 6–8, 10, 12, 15–18, 24, 33, 34

deme

Generic name for a single unit of space corresponding to a populated area. This can be assumed similar to a village. 11–14, 34

job scheduler

Application used to manage and launch a set of jobs, usually on multiple computers organised as nodes of a cluster, and regulating the amount of work each node is doing at every moment. 23

posterior

In the context of an ABC statistical analysis, a resulting parameter distribution. 18, 19, 27, 28, 36, 37

prior

In the context of an ABC statistical analysis, an input parameter distribution. 18, 19, 27, 36

Bibliography

Cox, M. P., Karafet, T. M., Lansing, J. S., Sudoyo, H., and Hammer, M. F. (2010). Autosomal and X-linked single nucleotide polymorphisms reveal a steep Asian–Melanesian ancestry cline in eastern Indonesia and a sex bias in admixture rates. *Proceedings of the Royal Society of London B: Biological Sciences*, 277(1687):1589–96.

Csillery, K., François, O., and Blum, M. G. B. (2012). abc: an R package for approximate Bayesian Computation (ABC). *Methods in Ecology and Evolution*, 3(3):475–479.

Csilléry, K., Blum, M. G., Gaggiotti, O. E., and François, O. (2010). Approximate bayesian computation (abc) in practice. *Trends in Ecology & Evolution*, 25(7):410 – 418.

Guillot, E. G., Hazelton, M. L., Karafet, T. M., Lansing, J. S., Sudoyo, H., and Cox, M. P. (2015). Relaxed observance of traditional marriage rules allows social connectivity without loss of genetic diversity. *Molecular Biology and Evolution*, in press.

Kehdy, F. S. G., Gouveia, M. H., Machado, M., Magalhães, W. C. S., Horimoto, A. R., Horta, B. L., Moreira, R. G., Leal, T. P., Scliar, M. O., Soares-Souza, G. B., Rodrigues-Soares, F., Araújo, G. S., Zamudio, R., Sant Anna, H. P., Santos, H. C., Duarte, N. E., Fiaccone, R. L., Figueiredo, C. A., Silva, T. M., Costa, G. N. O., Beleza, S., Berg, D. E., Cabrera, L., Debortoli, G., Duarte, D., Ghirotto, S., Gilman, R. H., Gonçalves, V. F., Marrero, A. R., Muniz, Y. C., Weissensteiner, H., Yeager, M., Rodrigues, L. C., Barreto, M. L., Lima-Costa, M. F., Pereira, A. C., Rodrigues, M. R., Tarazona-Santos, E., and Consortium, T. B. E. P. (2015). Origin and dynamics of admixture

in Brazilians and its effect on the pattern of deleterious mutations. *Proceedings of the National Academy of Sciences*, 112(28):8696–8701.

Lansing, J. S., Cox, M. P., de Vet, T. A., Downey, S. S., Hallmark, B., and Sudoyo, H. (2011). An ongoing Austronesian expansion in Island Southeast Asia. *Journal of Anthropological Archaeology*, 30(3):262 – 272.

North, M. J., Collier, N. T., Ozik, J., Tatara, E. R., Macal, C. M., Bragen, M., and Sydelko, P. (2013). Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1).

Smouse, P. E., Long, J. C., and Sokal, R. R. (1986). Multiple regression and correlation extensions of the Mantel test of matrix correspondence. *Systematic Zoology*, 35(4):627–632.

Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., and Dessimoz, C. (2013). Approximate Bayesian Computation. *PLoS Comput Biol*, 9(1):e1002803.

Weissgerber, T. L., Milic, N. M., Winham, S. J., and Garovic, V. D. (2015). Beyond bar and line graphs: Time for a new data presentation paradigm. *PLoS Biol*, 13(4):e1002128.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer: New York.

Appendix A.

Examples of visualisation

This appendix presents different visualisation examples that are generated by the developed scripts.

I. Stability

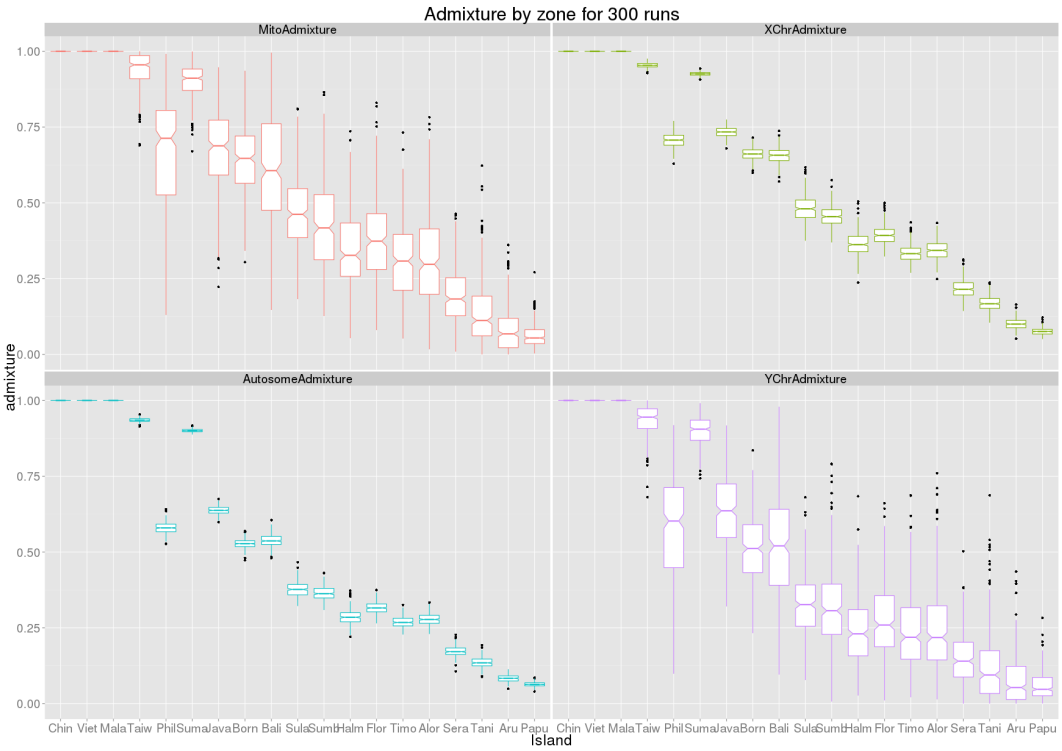


Figure A.1.: *Box-plots of admixture by island, separated by type of DNA.*

II. Sensitivity

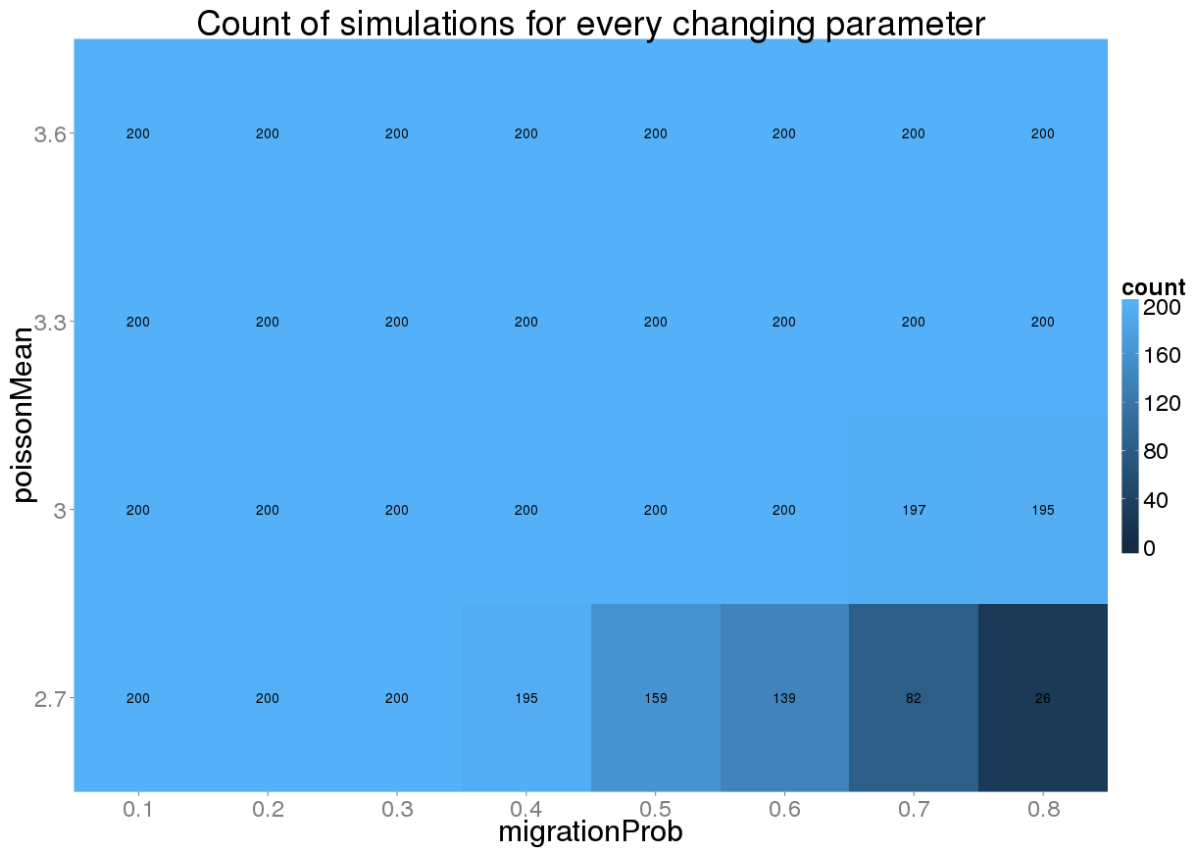


Figure A.2.: *Heat-map of counts of simulations for a one-dimensional sweep of migrationProb and poissonMean values. Failed simulations are revealed for both high migrationProb and low poissonMean values (dark boxes).*

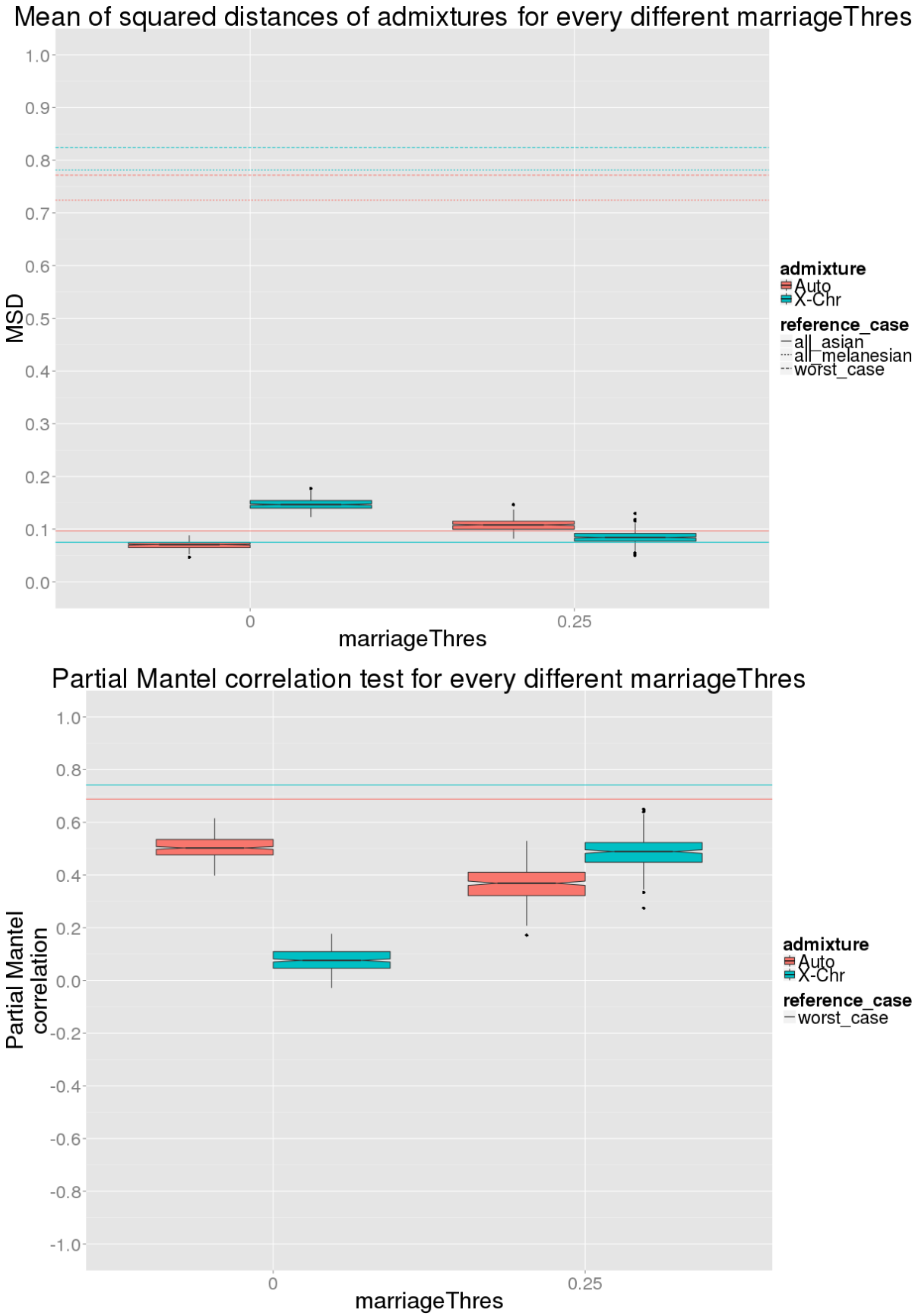


Figure A.3.: *Box-plots of comparisons of simulations vs. real data for a one-dimensional sweep of marriageThres values. Additional lines corresponding to pre-defined extreme reference cases.*

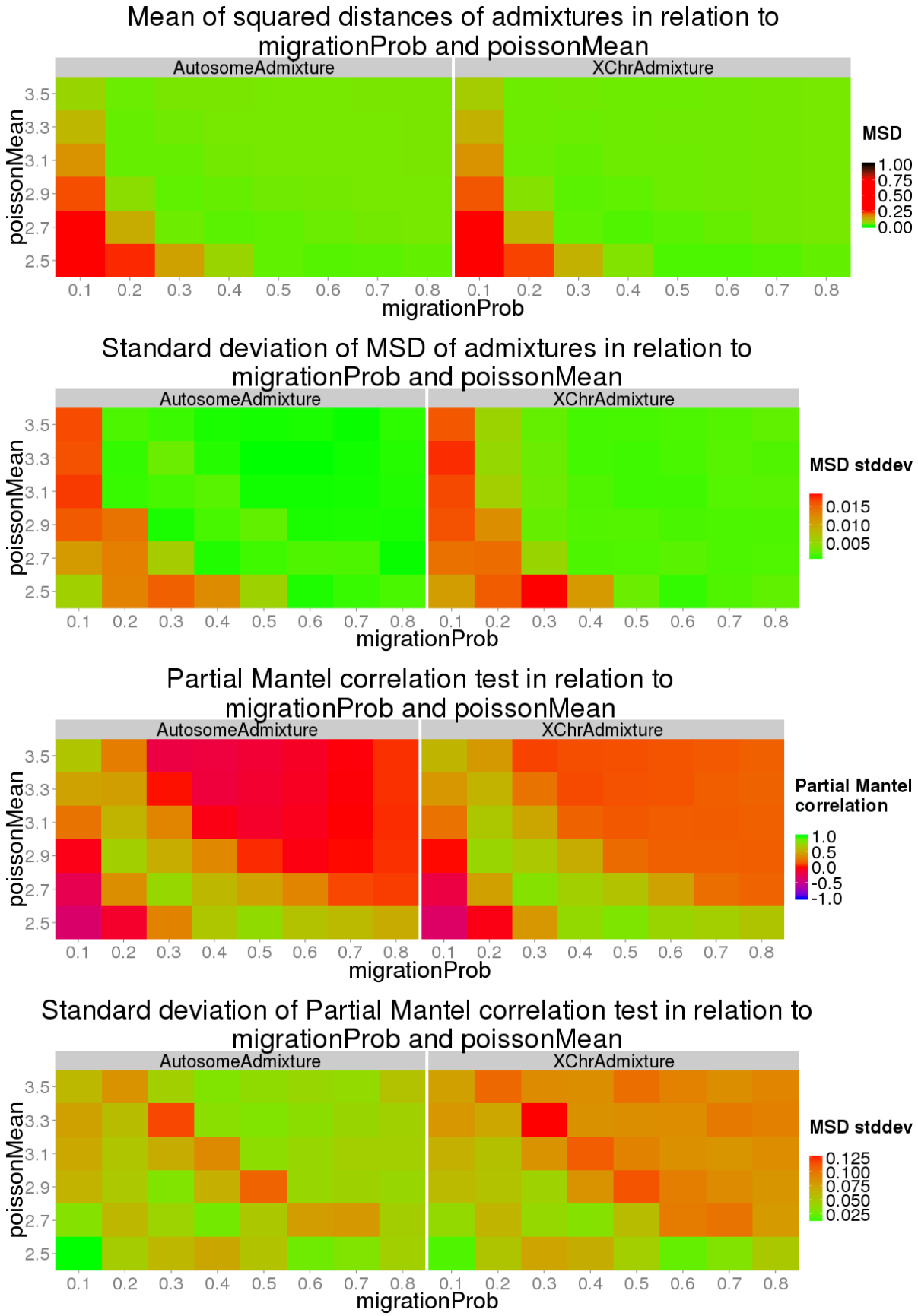


Figure A.4.: *Heat-maps of comparisons and corresponding standard deviations of simulations vs. real data for a two-dimensional sweep of migrationProb and poissonMean values.*

Appendix B.

Benchmarks

This appendix presents maximum memory and time usage at two points critical points in the pipeline (Figure 3.1).

I. Merging

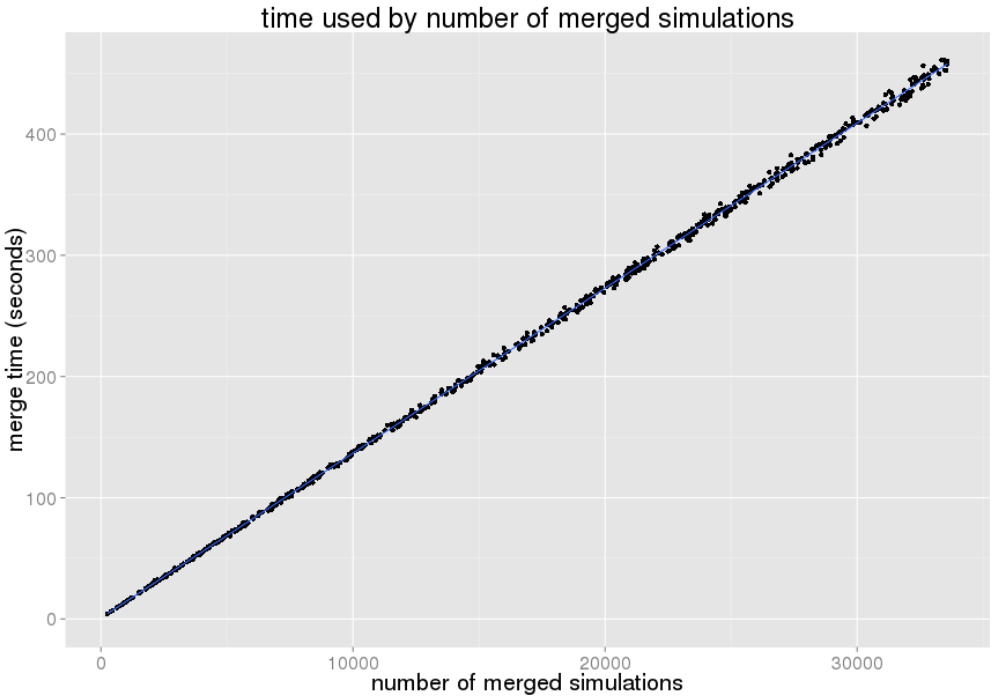


Figure B.1.: *Time used by the scripts in relation to the number of simulations merged. Corresponding, in Figure 3.1, to the* `merge` *script.*

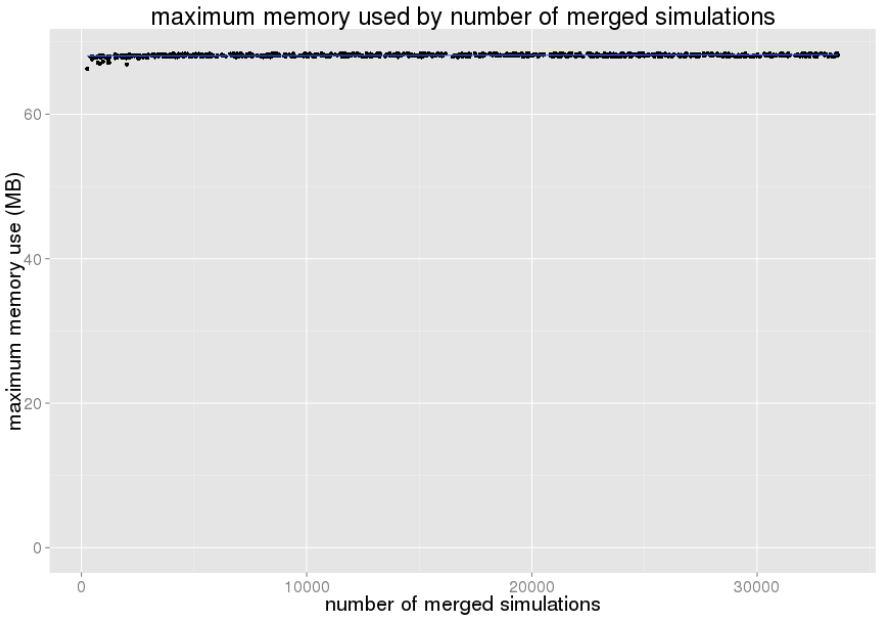


Figure B.2.: *Maximum memory used by the scripts in relation to the number of simulations merged. Corresponding, in Figure 3.1, to the* `merge` *script.*

II. Analysis

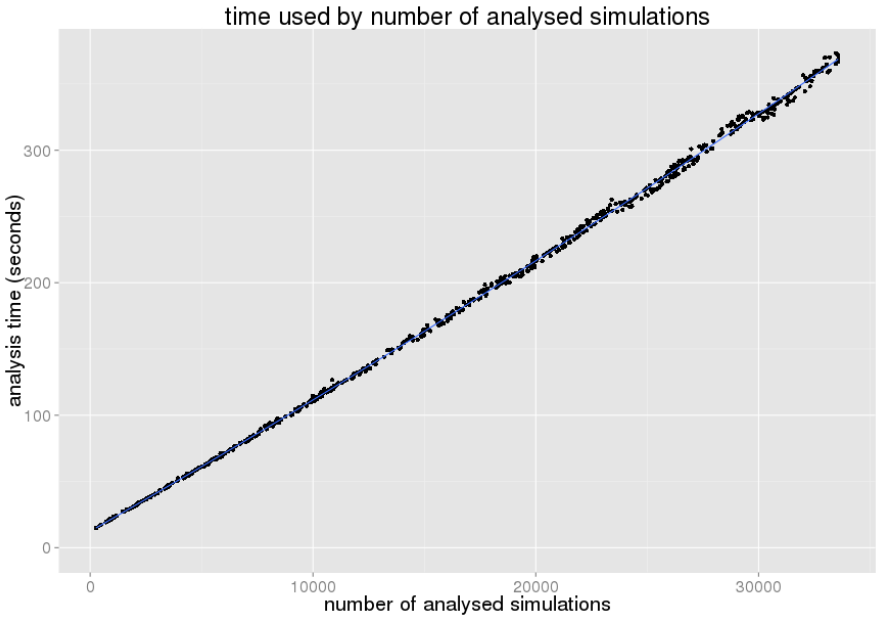


Figure B.3.: *Time used by the scripts in relation to the number of simulations analysed. Corresponding, in Figure 3.1, to the* `analysis` *and* `ABC` *scripts and to the creation of the corresponding output files.*

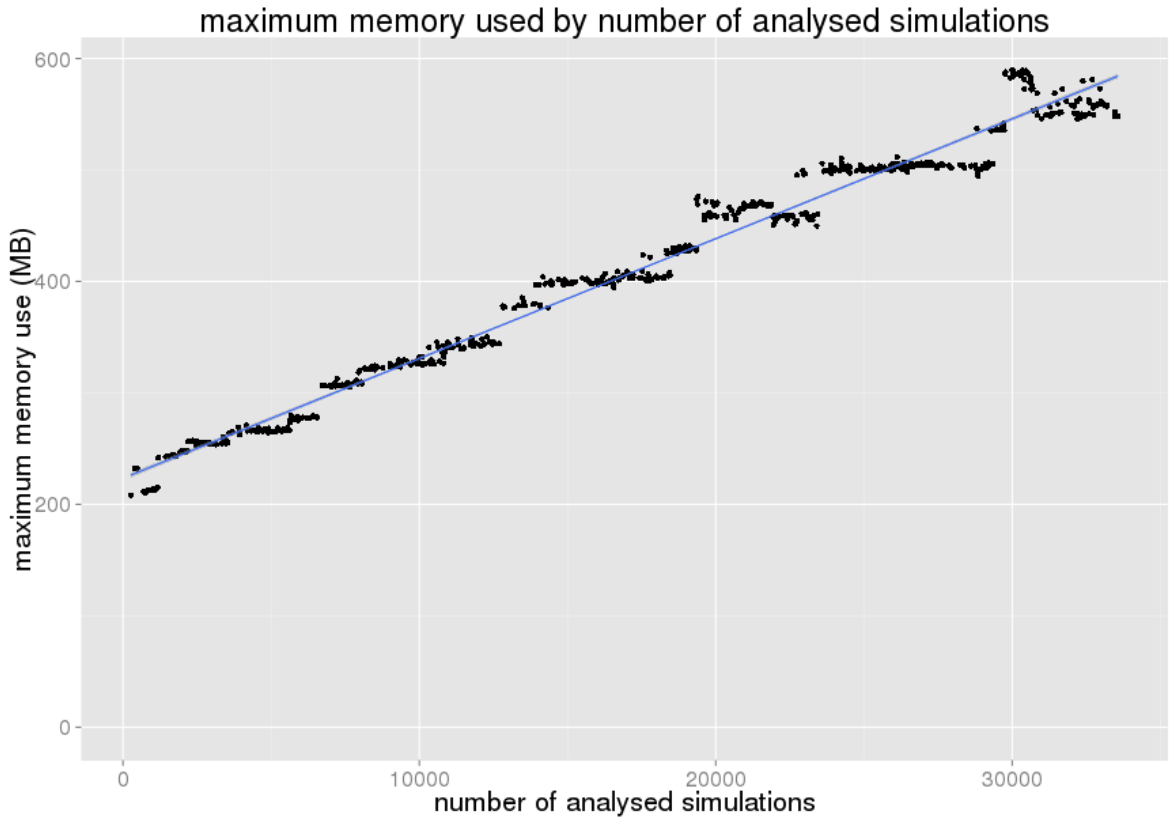


Figure B.4.: *Maximum memory used by the scripts in relation to the number of simulations analysed. Corresponding, in Figure 3.1, to the* `analysis` *and* `ABC` *scripts and to the creation of the corresponding output files.*