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

Year 2014-2015

Contents

Introduction

Two groups of populations can be identified in the Islands of South-East Asia (ISEA), one is composed of the Melanesians, whose ancestors settled in these islands during the first human settlement, 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 mainland China settled in the islands. This expansion implied great changes for the people living in the islands, including the development of the 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 an Asian ancestry or a Melanesian one. These markers are based on single nucleotid polymorphisms (SNPs) located in different chromosomes and 52 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 and allows to define the ancestry of a person 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.

Chapter 1

Project presentation

The project follows other studies made jointly at Massey University, the University of Arizona, the Santa Fe Institute, and the Eijkman Institute. These studies provide the set of SNPs used, determined after the sequencing of 1.430 individuals from 60 populations. The data and discoveries made led to further questioning, such as trying to explain the different admixture patterns, and this project tries to find answers to them.

This project consists of developing a model of the Austronesian expansion throughout the ISEA that could reproduce the same two patterns observed in the real data. The first pattern can be seen in the figure 1.1 and is a non-linear gradient in Asian admixture that declines abruptly around the eastern part of Indonesia. The second one, represented in the figure 1.2, shows the differences of admixtures on distinct parts of the DNA for different islands. These last differences would tend to appear in a context where marriages between Asian women and Melanesian men would be favoured.

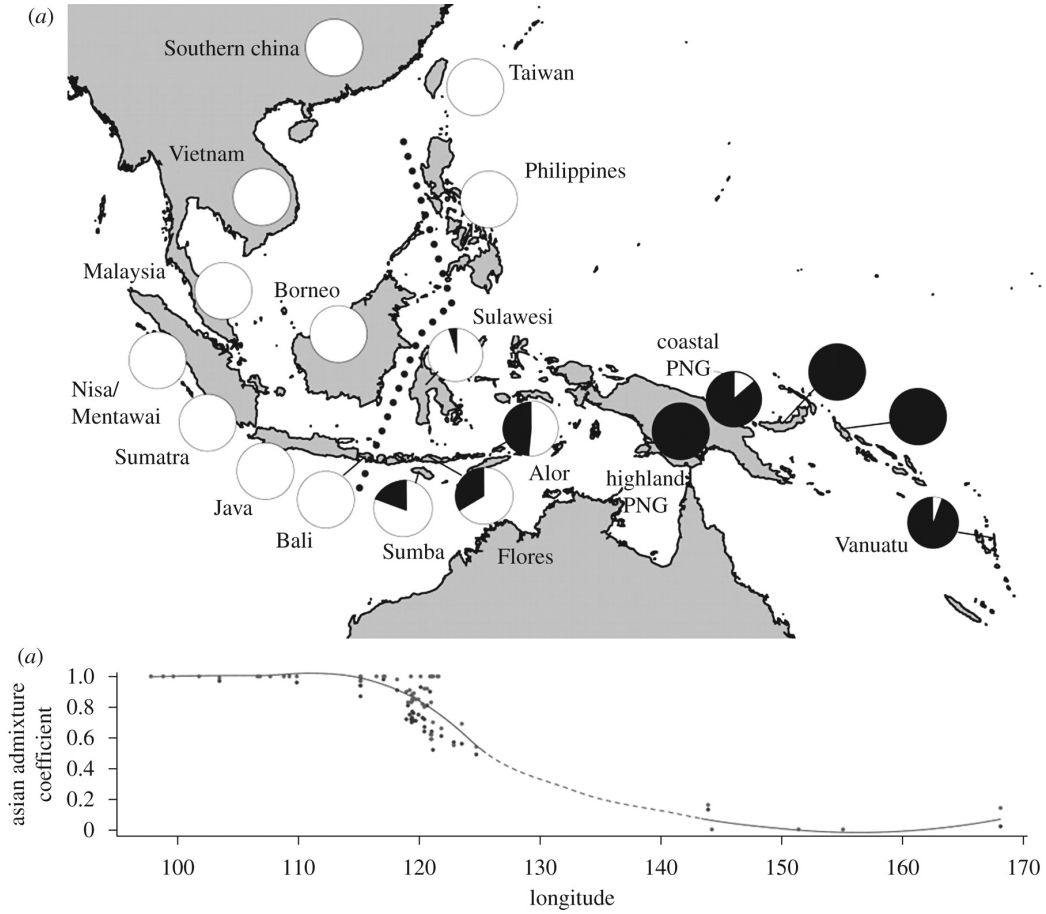


Figure 1.1: Local admixture rates across the Indo-Pacific region. (a) Pie charts showing mean regional admixture rates (Asian component in white; Melanesian component in black). Wallace's biogeographic line is shown as a dotted line. Regional admixture rates are shown for data reduction purposes; (b) 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].

The whole idea of the project is thus to try to reproduce the same patterns and to determine which sets of parameters led 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 nowadays.

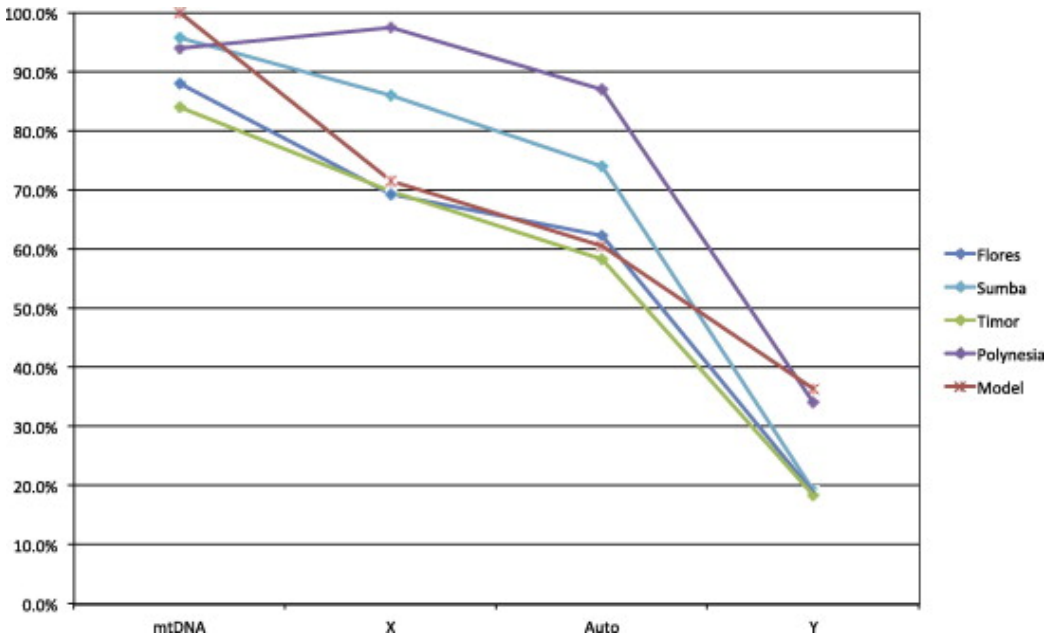


Figure 1.2: Fraction of Asian DNA in four genetic systems compared with model results for $\alpha = 0.02$ and 50 generations. Sample sizes: Flores = 453, Sumba = 639, Timor = 529. Reproduced unmodified from [Lansing et al., 2011].

1.1 Simulated data

The simulated data from the model will be the admixtures of the populations among the area covered by the model. The admixtures 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 for the last step of the simulation, since they will be the only values comparable to the real observed admixtures.

1.2 Agent-Based Model

The model will an Agent-Based Model that will comprise a graph of nodes, each node corresponding to a deme, and each edge corresponding to a possible migration route between demes (see figure 1.3). The agents of the model will be people living in the graph, in every node, and these agents will be able to migrate and to create families. The graph and the agents living in it will

respond to a set of basic rules regulated by parameters and the goal is then to observe the emerging behaviours arising from it.

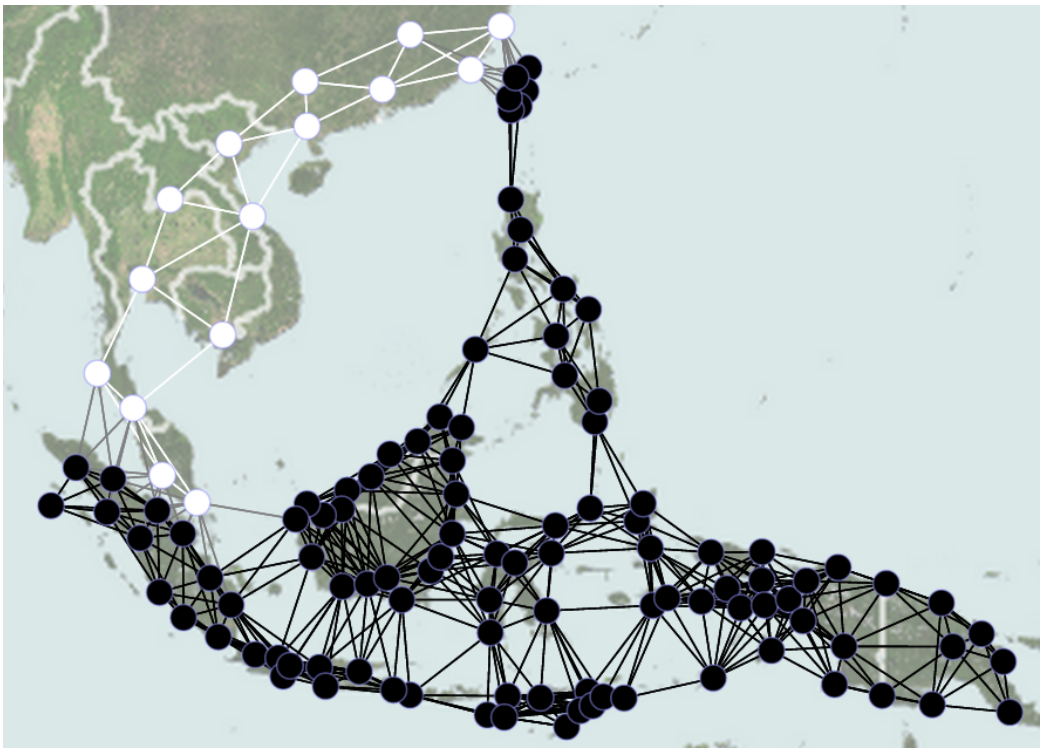


Figure 1.3: Nodes of a model superimposed onto the map of the modelled region. White nodes are the Asian nodes at the beginning of the simulation and black nodes are the Melanesian ones, according to one of the starting distributions used as parameter

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.

Other existing models

This model is one of the types of models that could have been chosen. For example, 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 the reality due to the higher time resolution.

Coalescent models also exist, they 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, they therefore don't have to simulate agents whose lineage would have ended up extinct. It is not really suited for this agent-based model though because parameters vary among agents and this is not trivial to do with coalescence.

Parameters

Numerous parameters can be set in the model. Some of them are defined once and for all and the fact that they are immutable is to simplify the actual use of the model by limiting the number of parameters to set. Changing them would actually not hold any meaning nor output realistic values.

The parameters that can actually be adjusted are listed in the table 1.1 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 corresponds to the topography of the graph used in the model, comprising the nodes and the edges between them, the latter is the distribution of the Melanesian and Asian populations in the graph at the initial step of the model.

The model can work with a lot of different parameter values, those corresponds to the "values" columns in the table 1.1, but knowing the context of

Parameter	Values	Estimated	Comment
Migration prob.	$\mathbb{R}_{0\leq x\leq 1}$	$\mathbb{R}_{0<x\leq 0.8}$	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	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{2.3<x<6}$	Poisson law mean for a Melanesian agent
Fecundity ratio	$\mathbb{R}_{\geq 0}$	$\mathbb{R}_{1\leq x\leq 1.5}$	corresponding ratio for an Asian agent
Marriage threshold	$\mathbb{R}_{0\leq x\leq 1}$	$\mathbb{R}_{0<x<0.5}$	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	$\mathbb{Z}_{\geq 0}$	$\mathbb{Z}_{50<x<400}$	pop. size in each deme, initially
Graph	{...}	{...}	composition of the graph (nodes and edges)
Starting distribution	{...}	{...}	distribution of pop. in the graph

Table 1.1: Summary of the changeable model parameters

the model, especially using previous anthropological studies, an estimation of possible realistic values can be found easily and will need to be refined later on during the analysis processes (see 2.4).

1.3 Statistical analysis framework

Given that the model will be run numerous times, with a lot of different changing values for the parameters, a robust statistical analysis framework needs to be implemented in order to be able to extract meaningful information from the high quantity of generated data.

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 the high quantity of data and the high dimensionality of it make difficult the use of standard statistical frameworks. It still relies highly on standard statistical tools but instead of giving one best result, it tries to give a distribution of best possible parameters. It answers the question of, given a set of parameter distributions, that are called "priors", what are the distributions, subsets of the priors, that give best results. This set of outputted distributions are then called "posteriors".

Its use has been increasing recently and some examples of recent papers using the ABC in similar contexts are [Guillot et al., 2015] or [Kehdy et al., 2015].

Chapter 2

Implementation

2.1 Overview

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 corresponds either to a single person or a couple that evolve in the demes. Each deme is a node in a graph and possible migration path between demes are edges of this graph. Therefore, the implementation of the model consists of agents evolving in a graph.

The different parts of the statistical analysis pipeline implementation are detailed in the next sections. A global overview of the pipeline is available in the figure 2.1 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 the figure 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 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 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.

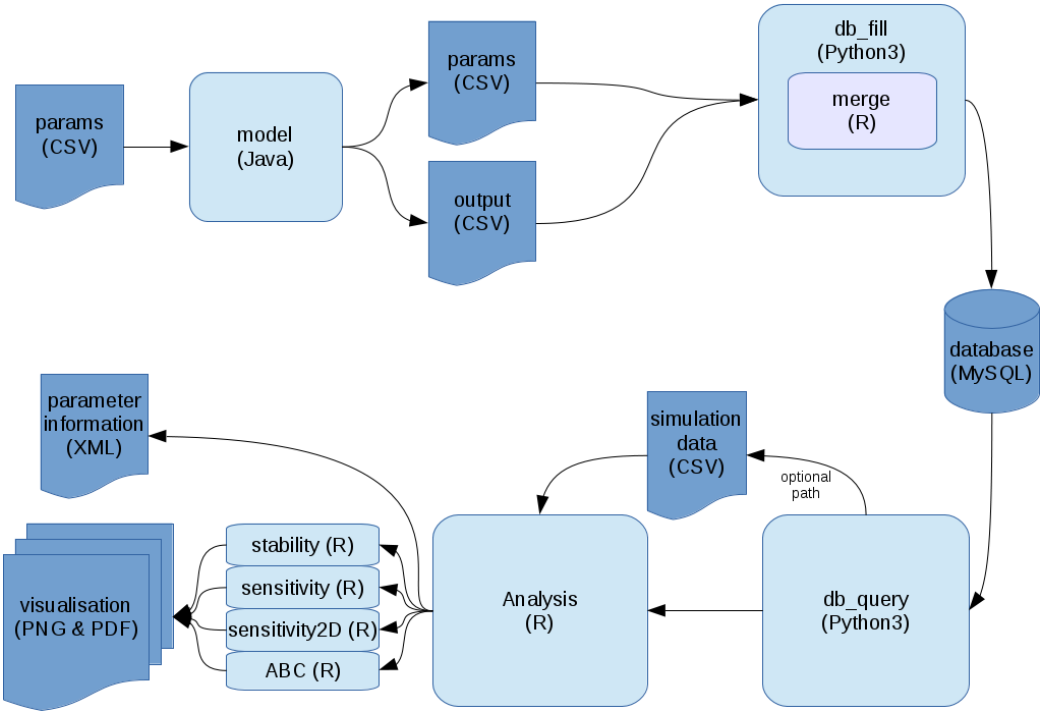


Figure 2.1: Visual representation of the different parts of the treatment and analysis pipeline. Dark colours represents data and the way it is stored and light colours represent programs and scripts and the language in which they have been written

2.2 Run management

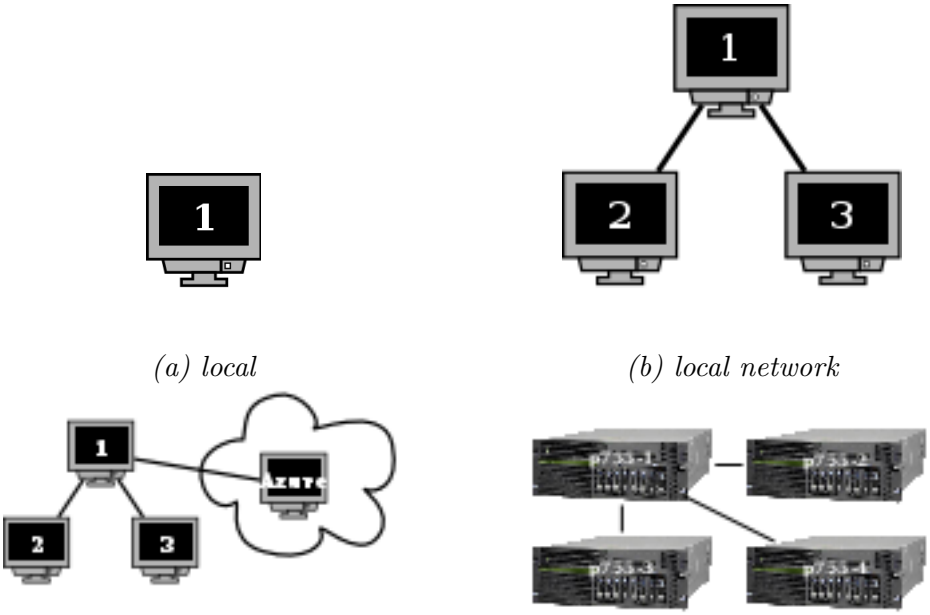
The fact that many simulations are required to be able to infer meaningful information from the model imply that they cannot be run sequentially on a single desktop computer in a reasonable amount of time. Luckily, every simulation being independent from the others, there are multiple ways to generate 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 2.2a), on a single computer, then using the three computers in the office as a cluster of compute nodes (figure 2.2b), for more heavy batches. Also, since Massey University just made an agreement with Microsoft Azure, that provides computing "in the cloud", simulations have been run transparently adding virtual machines on the Microsoft Azure system to the cluster of computers in the office (figure 2.2c). Finally, when the computation requirements were too high, for really huge batches, the NeSI[1]'s High Performance Computing (HPC) facilities have been used (figure 2.2d). 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 using 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 be a custom cluster of computers. The batches have to be submitted through a job scheduler, in this case the software is 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 of scenarios at the same time. In this case 4 nodes were used, each of those having 32 threads.

[1]New Zealand eScience Infrastructure



(a) local *(b) local network*

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

Figure 2.2: Different set-ups to run the model

2.3 Data processing, storage and query

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 to redo simulations for parameter values already tested and reducing the computation 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 the figure 2.3. 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.

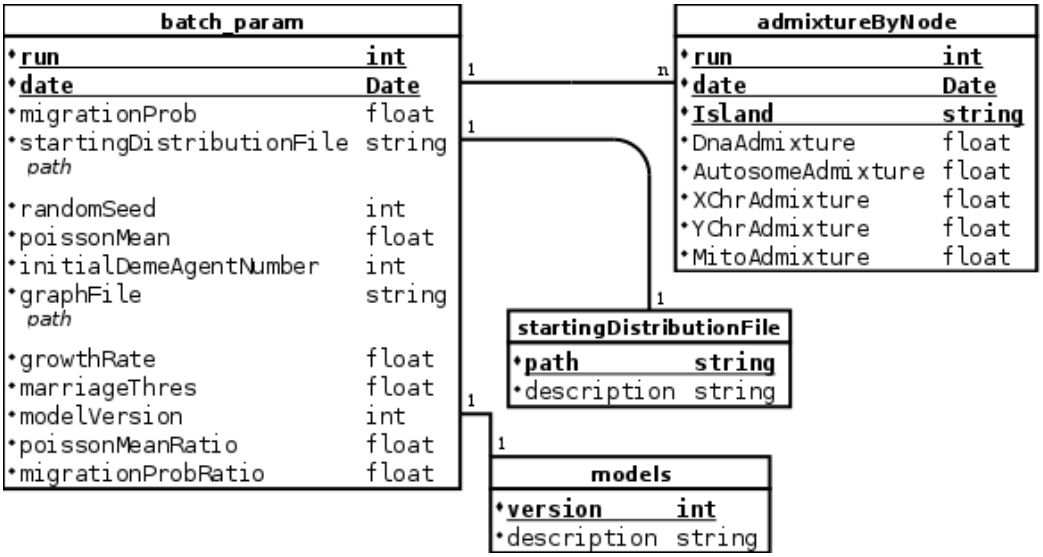


Figure 2.3: Structure of the database storing the results of the simulations and the corresponding parameters

The use of the database can be done through phpMyAdmin but two scripts have been developed to interact with it. The first one reads the two files outputted 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 user 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 an other script, in R, that processes the data before actually adding them. This way one process, the Python script, handles the connection with the database while an other process, the R script, does the actual data processing at the same time, in parallel.

2.4 Analysis

2.4.1 Comparisons

To be able to compare two different scenarios, comparison functions had to be defined. These functions will be able to provide a value of similarity

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 2.1: List of islands (or areas) in the model and in the reference data

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 to be of the same form than the observed data. Doing so allows them to be compared. The observed data does not include values for specific islands that, in change, are included in the simulations (see table 2.1). This is because nodes had to be placed in the model so that the agents could be able to migrate, even though the studies didn't 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.

The first, and most straightforward way two 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. An other measure was needed that would keep this information and take it into account while comparing the data. Also, the idea came to also compare the whole matrix of differences of admixtures among islands instead of the values island by island.

16 comparison functions were tested with randomly generated data. These

data had the structure of the 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 chose two functions whose results are highly correlated because it would mean to add dimensions to the comparison values while not actually adding any meaningful information.

Two of them have been selected that can hold different information about a comparison. The mean square distance (MSD) and the partial Mantel test have been selected as 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 higher possible MSD value will be around 0.8 (see reference cases lines in the upper part of figure A.4).

Partial Mantel test

The Mantel tests have been developed to be able to compare two matrices with the same information. The partial Mantel test is one of them that also

uses a third matrix, holding geographical distances of the cells of the matrices 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 matrices contain the values of distances of admixtures between every islands in the graph and a matrix M is calculated as

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

where d is the function returning the distance between the two arguments and Ad the admixtures of the n islands of the graph. With the corresponding matrices for the simulation data, the real data and also the geographical distances, the partial Mantel test can be done as such

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

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.

2.4.2 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 it well so that it is useful to the user. 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 this data. Whenever possible, notched box-plots were used to really try to

show the distributions and shapes of the data. The box-plot 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 much box-plots would overwhelm the user so, as an alternative, mean values are used but with additional error bars corresponding to the standard deviation in order to be able to evaluate quickly 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. The figure 2.4 shows a part of an example visualisation showing both good and bad correlation values for the autosomal and X-chromosomal admixtures.

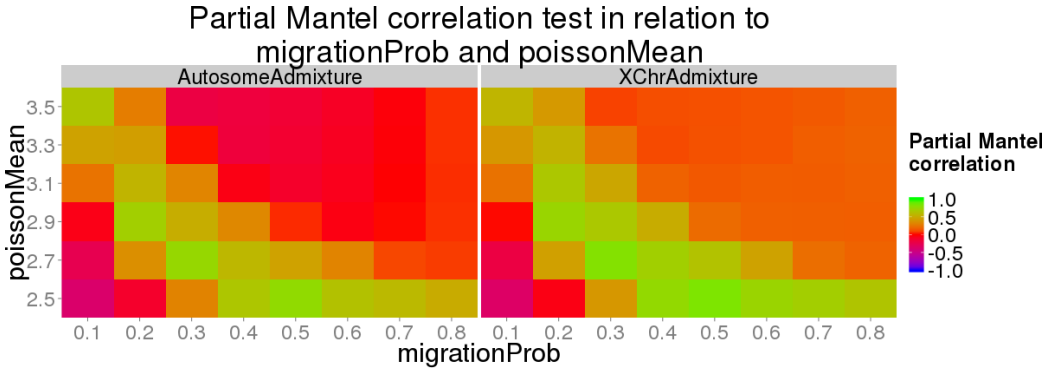


Figure 2.4: View of a part of an outputted graph, highlighting the different possible colours in the visualisation. The complete graph is in the appendices, figure A.5.

Examples are available in the appendix (see 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 3.

Also, a common theme have been defined for all the visualisations to be able 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). This vectorial output allows to apply changes to the images for publication if a specific ratio is required or if changes need to be made to any part of the image, but more importantly, this is the only way to have a lossless format that can be resized without quality loss.

Chapter 3

Results

3.1 Grid search analysis

In order to reduce the parameter space, a grid search has been done. It consists of going through the possible parameter sets by setting parameter values at regular steps throughout the space and to run n multiple simulations for every point in the space in order to have statistically meaningful results for every point. One hypothetical parameter grid search can be seen in figure 3.1.

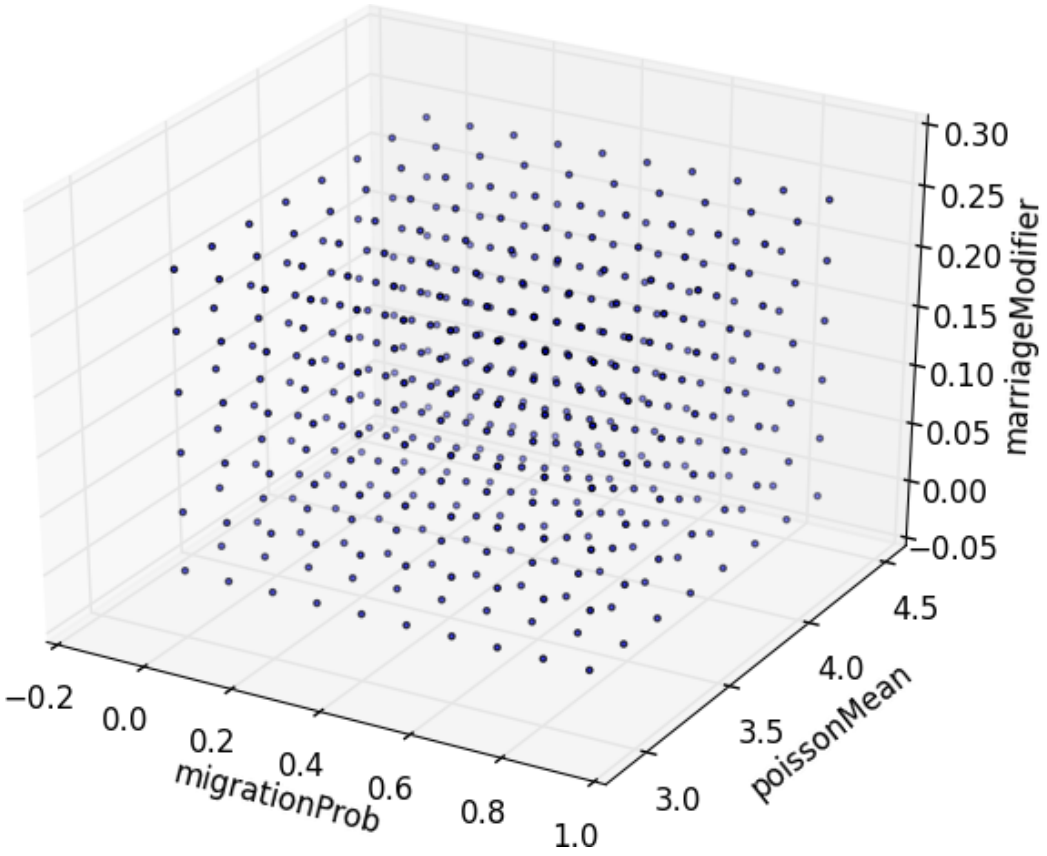


Figure 3.1: Visual representation of a grid search on three parameters, $migrationProb$, $poissonMean$ and $marriageModifier$. Each point represents a set of those parameters, that will be used for n different simulations

Depending on the number of parameters and the size of the value steps, this holds 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.

Stability analysis

There was multiple ways to analyse outputs from simulations with grid searches. 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 on the data for this point in the grid. The first ones are 4 graphs, one for each type of observed DNA, wherein box-plots for the admixture values of each island are displayed (partial view in fig. 3.2, 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 admixtures are supposed to be less stable in the contact zones between the different populations than in the extremities of the graph, where the source populations live. Less stability is also expected when looking at admixture values comprising less markers (mitochondrial and Y-chromosomal admixtures). And finally, lower stability is expected from islands that include less demes, and thus lower population, like Alor, Aru or Bali that are composed of only one single deme each. Any other behaviour leads to believe that the parameter set it comes from is not realistic, thus inducing abnormal model behaviours.

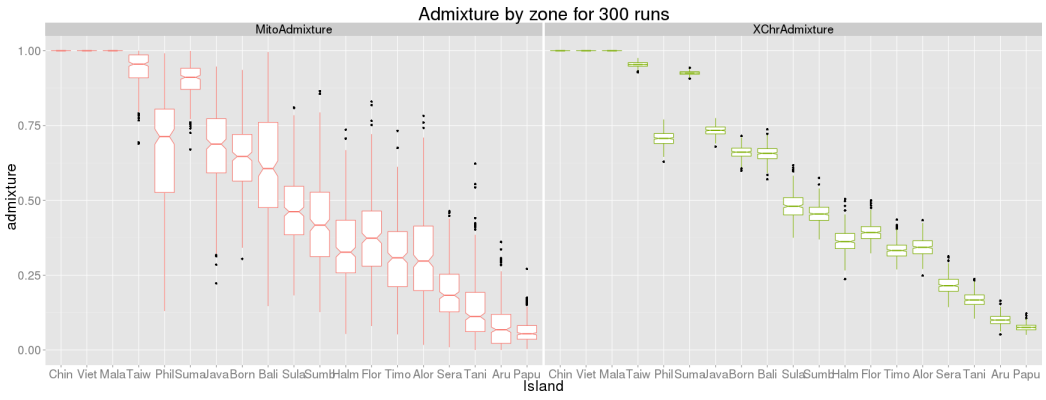


Figure 3.2: View of a part of an outputted graph, showing box-plots for mitochondrial and X-chromosomal admixture values by island. The complete figure is in the appendices, figure A.1.

An other view that is outputted by the stability analysis is the graph

of admixture values as a function of the types of DNA, and this for every island (fig. A.2). This graph aims at resembling the figure 1.2 from [Lansing et al., 2011] as a way to compare the real data in this paper to the data generated in the simulations.

Sensitivity analysis

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

At first, this analysis was done by looking at a single parameter changing while the others were fixed at one value, this was called a one-dimensional sweep and corresponds to parameters 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 to further visualise 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, that is why a subset of the simulations were used when doing the visualisation part.

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 not comparable. An example of this visualisation can be seen in the figure A.3.

Then, comparisons to the real data are done, for both the Autosomal and X-Chromosomal DNA data (see 2.4.1) and the resulting values are displayed so they can be analysed by the user as seen in figures A.4 and A.5.

3.2 ABC Framework

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

At first, the R package "ABC" [Csillery et al., 2012] has been 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 has been 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.

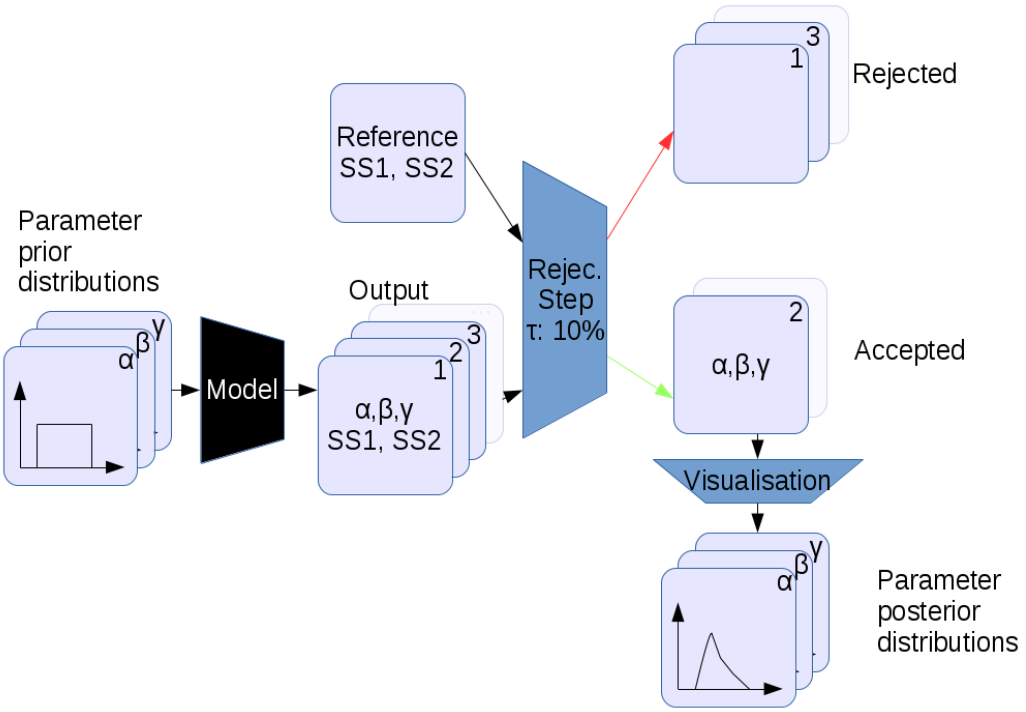


Figure 3.3: Visual representation of the different steps of an ABC inference framework

The whole framework encompasses the steps from the choice of the parameter sets to the inference of the posterior distributions. The figure 3.3 summarises the successive steps of this process with hypothetical parameters α, β and γ and summary statistics $SS1$ and $SS2$. In this project, the parameters used are the ones defined in the table 1.1 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 4 results of the comparison functions defined in the section 2.4.1. These are the MSD and the correlation for both the X-chromosome and for the autosomes.

3.2.1 Detailed steps of the ABC

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. 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 the previous grid search analysis). A visualisation of example simulation parameter sets that can be generated by the first step of an ABC framework can be seen in the figure 3.4.

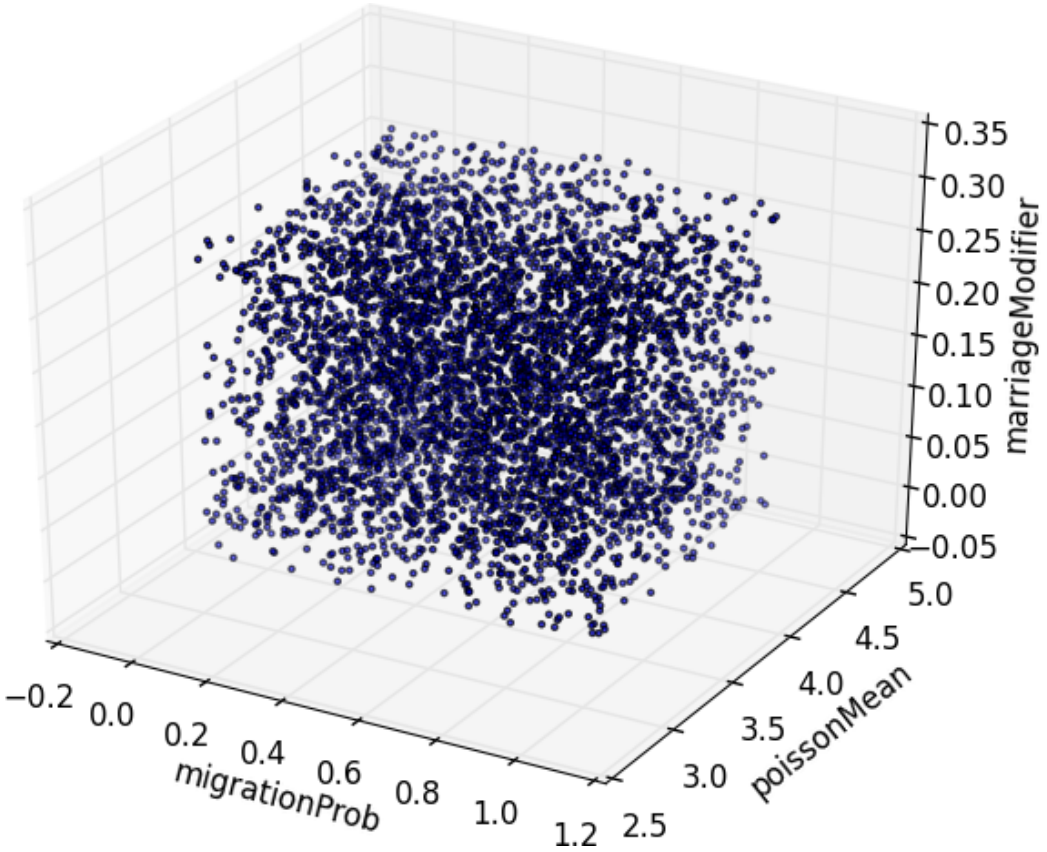


Figure 3.4: Visual representation of 5.000 sets of three parameters, $migrationProb$, $poissonMean$ and $marriageModifier$ drawn from independent uniform distributions, as a result the points are evenly spread inside a cuboid. Each point represents one set of parameters that will be used for one single simulation

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 have been biased.

After the model has generated results for every parameter set, the ABC framework needs to choose which simulations are the ones that are the 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 is expected to give "bad" results. The tolerance can be adjusted for every analysis and no perfect tolerance exists, this has to be defined by the analyst using the ABC.

After that, the parameter values for the selected simulations are retrieved and the corresponding distributions correspond to 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 reality.

Chapter 4

Discussion

4.1 Importance of randomness

One important aspect that has been discovered while working with the Repast framework was 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 the same one.

When looking at the results with 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 to an other simulation. 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 bits signed integer, meaning that there are more that 4 billions possibilities,

collisions happened more than once during this project.

The first way to handle this has been to alert the user when this happened, letting him 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 collision.

4.2 Optimisations

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 really 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 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 the appendix B in the graphs B.1 for the treatment step and B.3 for the analysis step. The corresponding maximum memory use values are in the graphs B.2 and B.4.

There is effectively still a great amount of room to improve in memory usage, especially seeing that the base memory usage is between 60 and 200 MiB just when loading libraries, functions, global variables and set-up code.

The stream approach also allows the different steps of an analysis to be run simultaneously, by piping each step to the next, 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.

4.3 Other visualizations

The visualisations done were made and adapted on the run 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 and other useful ones. They added a layer of abstraction that could have added performance problems but the choice of already mainly used ones limited this and also made it easier to evolve even though the person writing the code changed. 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 integrating 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.

Conclusion

The model used until is still undergoing changes, with the will to make it more accurate and to take into account more factors. The problem of adding complexity would be to add enough complexity to keep the model relevant while still being computationally simple enough so that it could be run in an acceptable amount of time. Several aspects can be changed, ranging from the structure of the family and how it is managed in the model, to higher level things 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 outputted 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 been found. Even if research needs to be followed-up, the first findings have been able to discard some scenarios and to confirm some other. 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.

This internship has been the 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 in order 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 has 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

Glossary

admixture Here, genetic admixture. Introduction of new genetic lineage into a population. Refers to the proportion of the genome coming from one ancestral population or an other. 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 of zero for a person having absolutely no Asian SNP, and since only two populations are considered, completely Melanesian. 2, 3, 5, 6, 15–17, 21

deme Generic name for a single unit of space corresponding to a populated area, this can be assumed similar to a village. 5–8, 10, 21

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. 12

posterior In a context of an ABC statistical analysis, a resulting parameter distribution. 8

prior In a context of an ABC statistical analysis, a input parameter distribution. 8

Bibliography

[Cox et al., 2010] 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*.

[Csillery et al., 2012] Csillery, K., Francois, O., and Blum, M. G. B. (2012). abc: an r package for approximate bayesian computation(abc). *Methods in Ecology and Evolution*.

[Csilléry et al., 2010] 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 et al., 2015] 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*.

[Kehdy et al., 2015] 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 et al., 2011] 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 et al., 2013] 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 et al., 1986] 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):pp. 627–632.

[Sunnåker et al., 2013] 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 et al., 2015] 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, 2009] 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

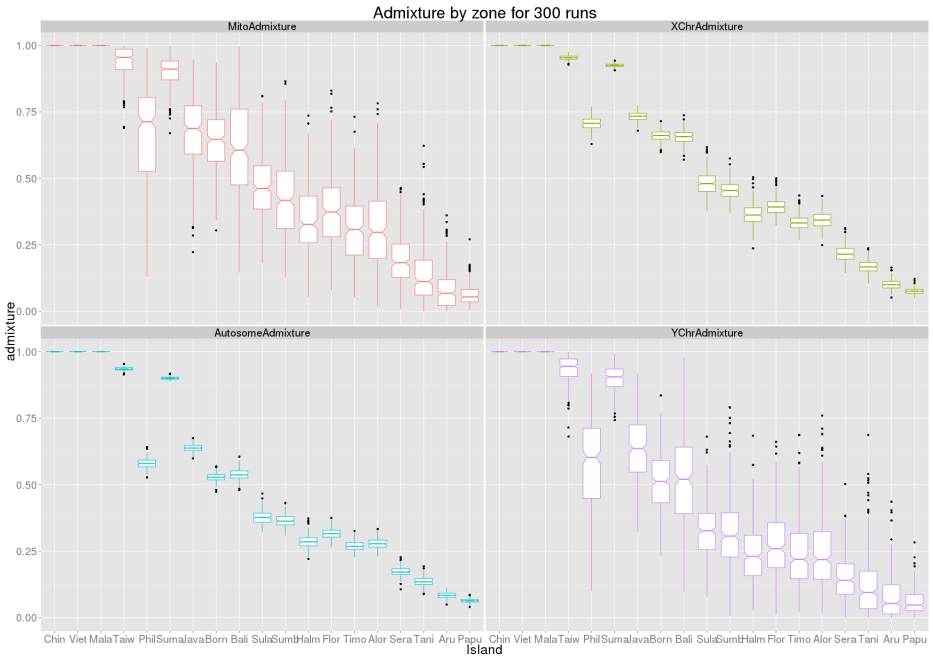


Figure A.1: box-plots of admixtures by island, separated by type of DNA

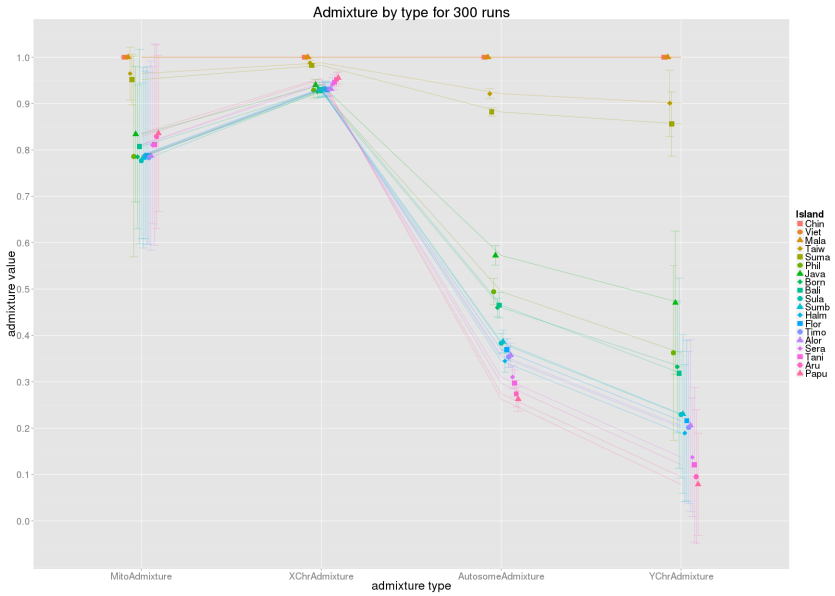


Figure A.2: admixture values by type of DNA, for every island

II Sensitivity

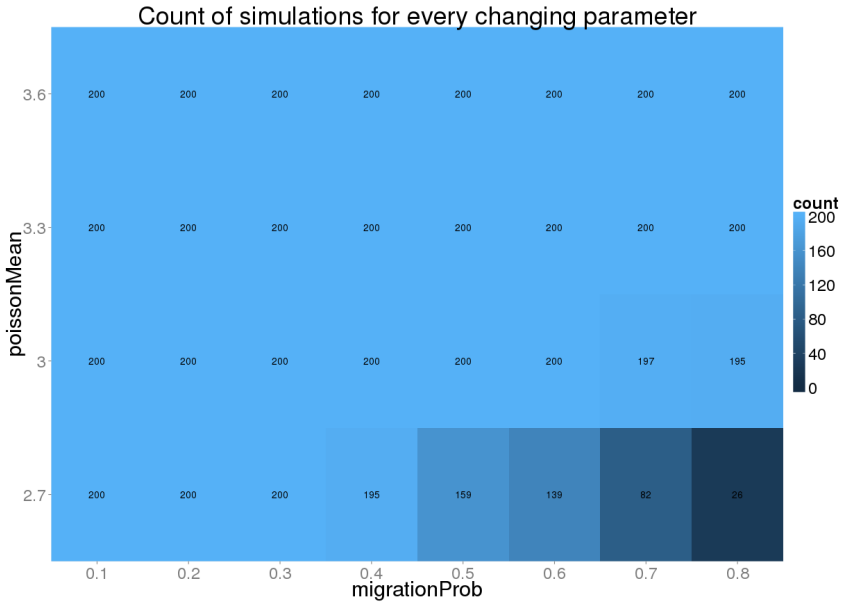


Figure A.3: Heat-map of counts of simulations for a one-dimensional sweep of migrationProb and poissonMean values. Failed simulations are revealed for both high migrationProb and low poissonMean values

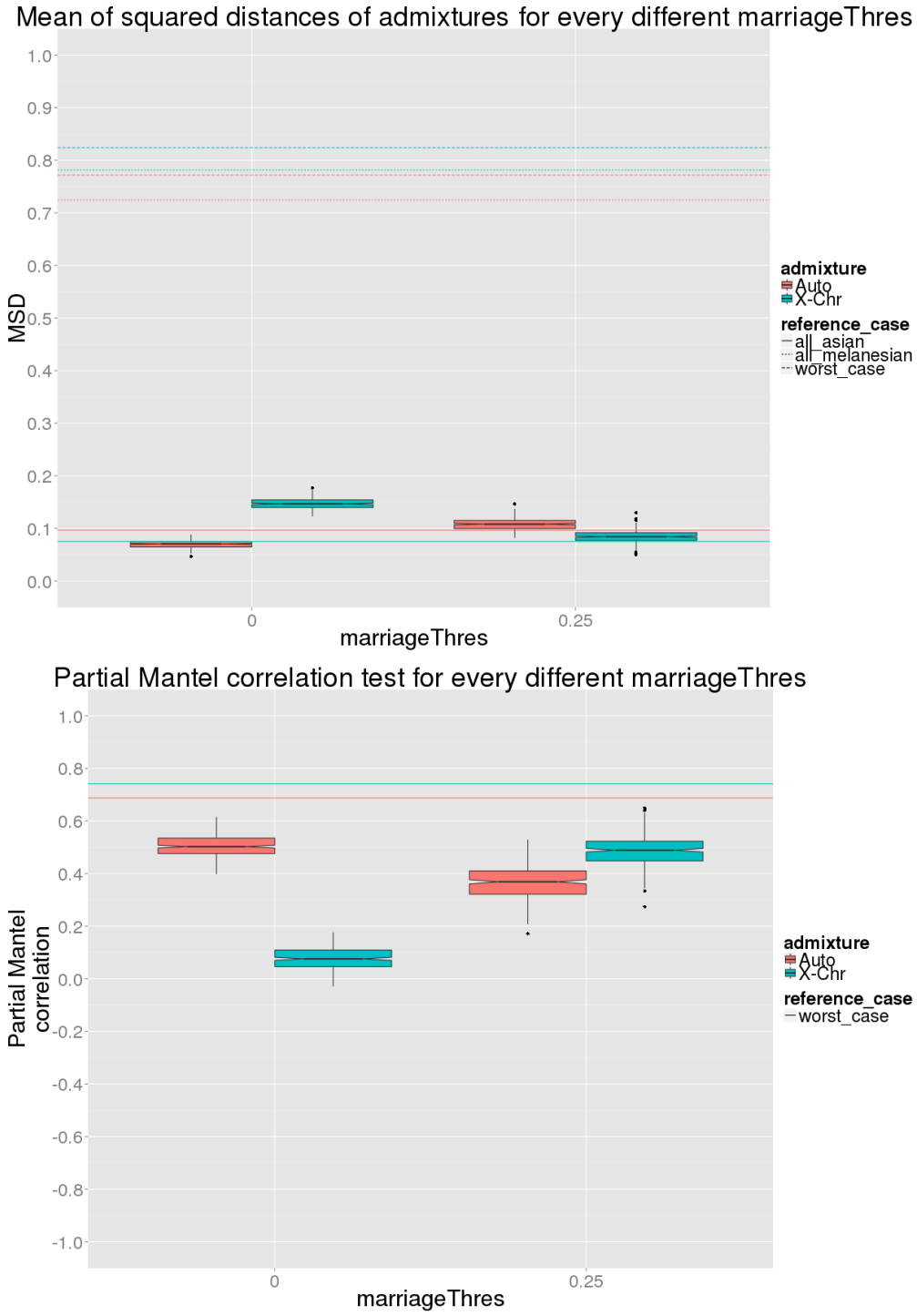


Figure A.4: Box-plots of comparisons of simulations vs. real data for a one-dimensional sweep of marriageThres values. Additional lines corresponding to predefined extreme reference cases

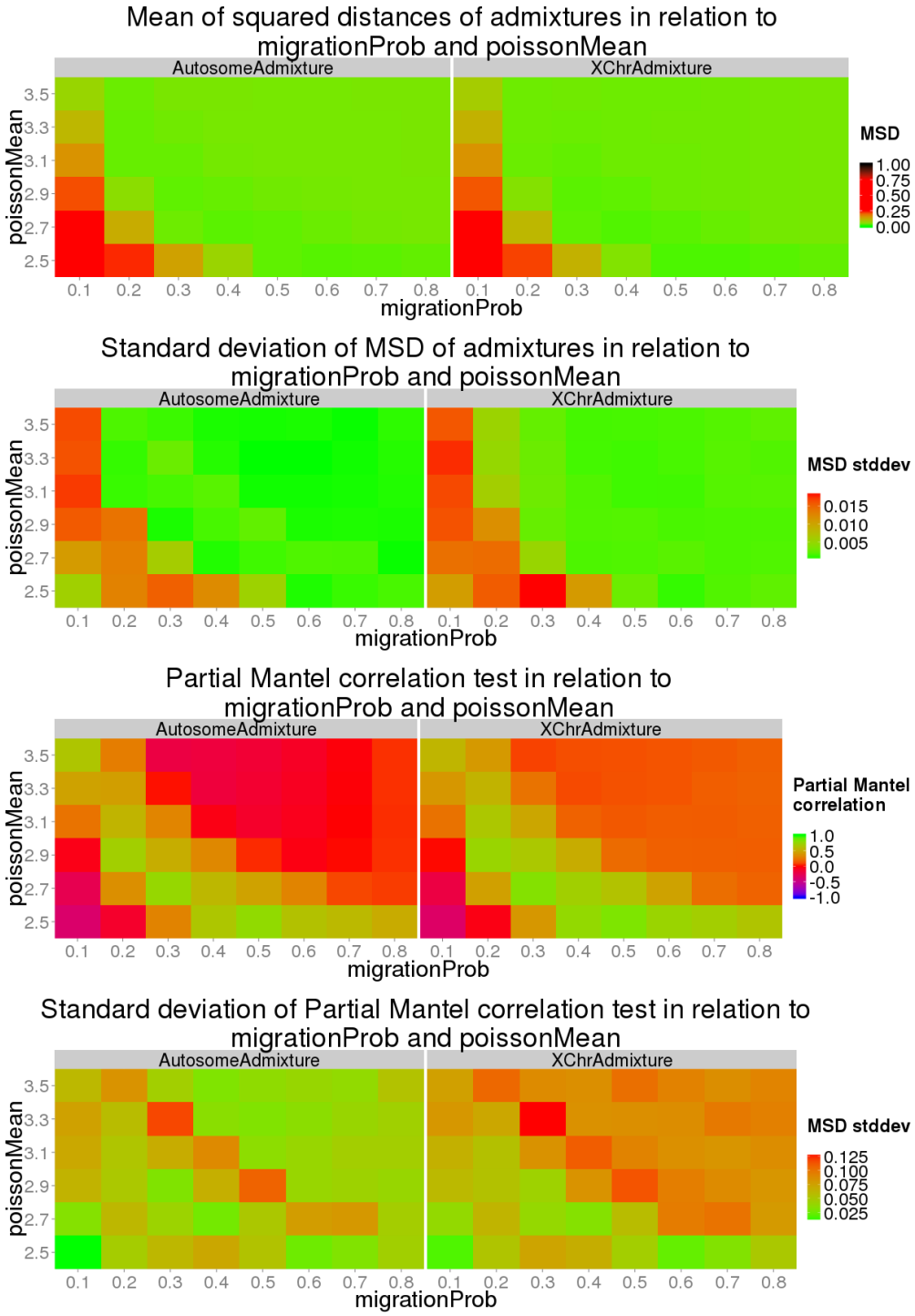


Figure A.5: Heat-maps of comparisons and corresponding standard deviations of simulations vs. real data for a two-dimensional sweep of migrationProb and poissonMean values

Appendix B

Benchmarks

I Merging

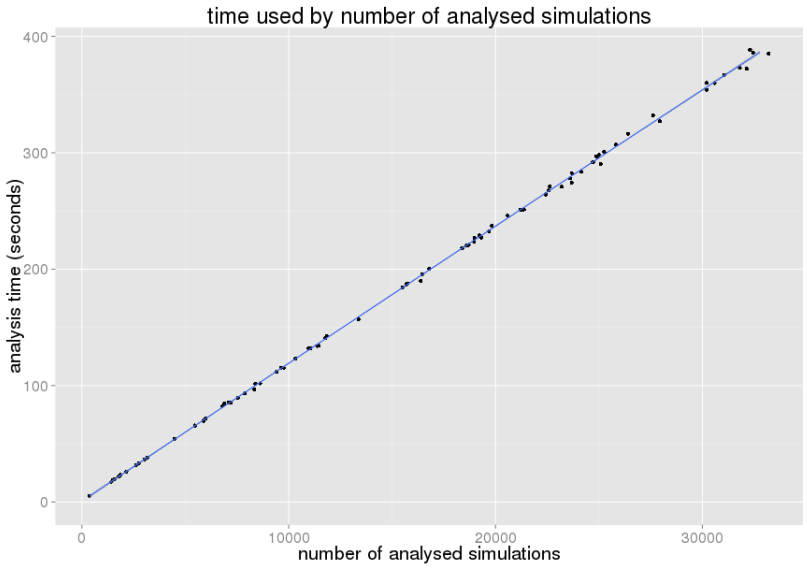


Figure B.1: Time used by the scripts in relation to the number of simulations merged. Corresponding, in the work-flow, to `merge.R`

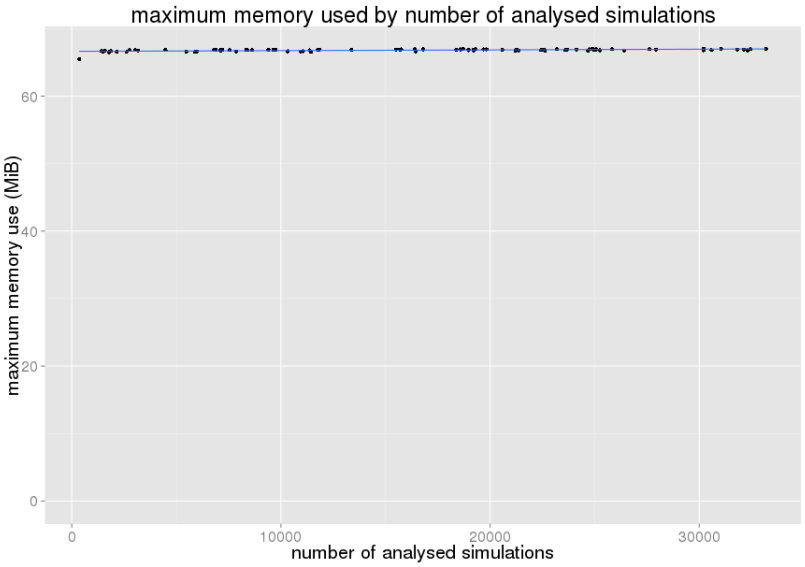


Figure B.2: Maximum memory used by the scripts in relation to the number of simulations analysed. Corresponding, in the work-flow, to `merge.R`

II Analysis

note: update with latest changes (hopefully, better values)

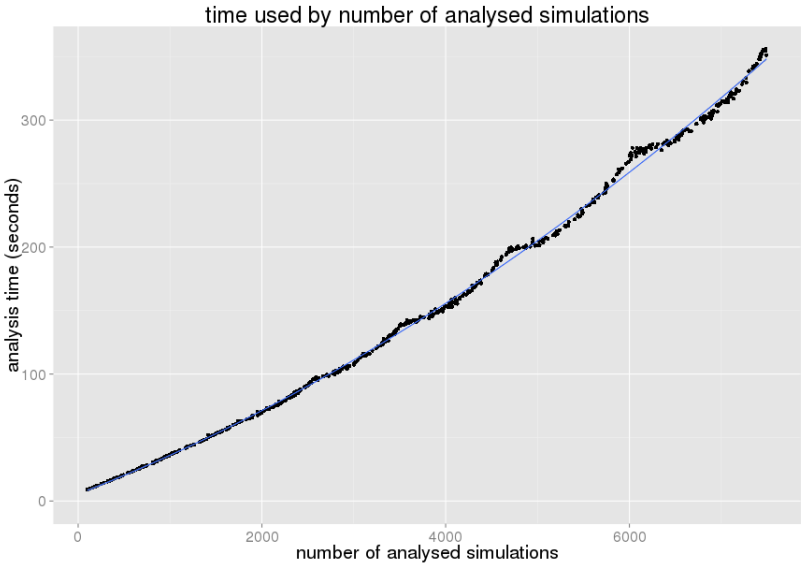


Figure B.3: Time used by the scripts in relation to the number of simulations analysed. Corresponding, in the work-flow, to `analysis.R`*,* `sensitivity.R` *and the creation of the corresponding output files*

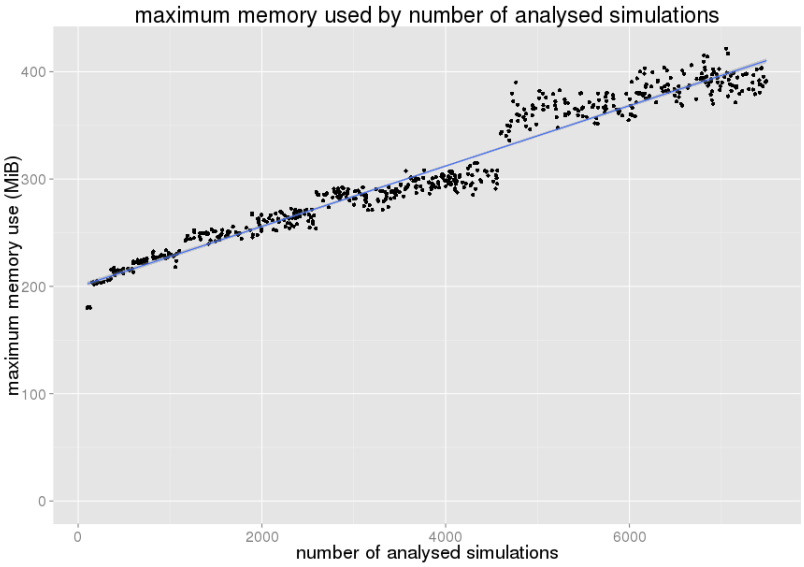


Figure B.4: Maximum memory used by the scripts in relation to the number of simulations analysed. Corresponding, in the work-flow, to `analysis.R`*,* `sensitivity.R` *and the creation of the corresponding output files*