UVIC
UNIVERSITAT DE VIC
UNIVERSITAT CENTRAL
DE CATALUNYA

Master of Science in Omics Data Analysis

Master Thesis

Evaluation of the performance of commonly applied global ancestry algorithms in complex spatial demographic scenarios

By

Roger Roig

Supervisor: Oscar Lao
Population Genomics Team Leader
Centre Nacional d'Anàlisi Genòmica (CNAG-CRG)
Centre de Regulació Genòmica

Department of Systems Biology
University of Vic – Central University of Catalonia

September 2016

ABSTRACT

The development of new methods for inferring ancestral origins in human populations has attracted a renewed interest for human population geneticists for better understanding recent human evolutionary history or for correcting the presence of hidden population substructure in genome-wide association studies (GWAS). The algorithms for detecting population substructure present several problems such as the dependency on the assumptions of the algorithm, the type and number of considered DNA markers, the underlying demographic relationship among the considered populations and the sample size of the target populations.

With this concern in mind, we have constructed an experimental model for testing the performance of currently algorithms applied for estimating population substructure which starts by designing two ideal prototypes of spatially structured populations (2D stepping stone and anisotropic). From each model we have generated a pool of 78 experimental datasets, simulating the genomic molecular diversity with Fastsimcoal2 under various migration rate conditions, performing the sampling of individuals and populations and selecting different filtering strategies: Minor Allele Frequency (MAF) and Linkage Disequilibrium (LD). Those 78 datasets (plink bed files) have been processed to evaluate the response of commonly applied algorithms to SNP data for quantifying individual population substructure: Principal Components Analysis (smartPCA), Multidimensional Scaling (MDS-PLINK), Spatial Ancestry Analysis (SPA), ADMIXTURE and SNMF. For those algorithms in which the output is a coordinate (PCA, MDS and SPA), we have evaluated the correlation (via Mantel and Procrustes tests) of these estimated coordinates with the geographic sampling coordinates of individuals in our original ideal artifacts. For ADMIXTURE and SNMF we have applied different algorithms for assessing the best K number of ancestries and we have applied CLUMPP software to compare their output matrices.

This ideal prototype has enabled us to establish the robustness of the five algorithms, identify best performing algorithms and determine the impact of the conditions imposed on the results of these programs.

TABLE OF CONTENTS

1. Introduction

1.1 Origin and maintenance of the genetic variation.

The genome is subject to permanent change over the evolution of a species and that is why we can not consider it as an immutable entity. Mutations (replications errors) are introduced by the cellular replication machinery during DNA replication at any cell and they are vigorously but not fully offset by the high fidelity of DNA polymerases and the DNA repair mechanisms. Henceforth, mutations are the ultimate origin of all genetic variation. However, taking into account just evolutionary consequences of mutation, we can follow only those changes that occur in the germline, and not those in somatic tissues because they are not heritable.

From a structural point of view, mutations comprise from simple nucleotide changes (called single nucleotide variant or SNV) to duplicating/deleting large fragments of the genome, as well as changes in orientation and genomic rearrangements in new genomic positions. Of all these possible types of mutations, the most common are SNVs. In addition to mutations, another physical factor that alters the genomic composition of variation is recombination. Meiotic recombination occurs as a part of sexual reproduction, and enhances the ability of populations to adapt to their environments by combining advantageous alleles at different physically contiguous loci. While alleles at loci on different chromosomes are randomly segregated during meiosis, alleles at loci closely linked on the same chromosome are not, as recombination between them occurs infrequently. Recombination can be studied at the population level by investigating whether specific alleles at different loci are correlated with one another more or less often than would be expected by chance. This nonrandom correlation is known as linkage disequilibrium (LD). The simplest model of recombination is that the rate of recombination is uniform. In other words, the probability of a crossover occurring between a pair of sequence variants is determined only by the physical distance that separates them. The products of this type of recombination event are two new haplotypes containing contiguous stretches of alleles from each ancestral haplotype.

Once a mutation in the germline has passed to the next generation, different evolutionary factors shape its frequency in the population. These evolutionary factors can be classified in demographic and selective.

Selective factors

From an evolutionary point of view and considering a very simplistic model of selection, there are two possible final scenarios for new mutations: if the new mutation provides a higher fitness to the carrier compared to the rest of individuals, the new mutation will increase its frequency in the population and, ultimately, achieve fixation. In contrast, if the new mutation provides a lower fitness to the carrier then it will be disadvantageous and removed from the population. Therefore, new mutations that modify the phenotype of an individual are the substrate of natural selection.

Obviously, much more complex evolutionary patterns exist in nature (i.e. multiple genes contributing to a phenotype, ancient ongoing balancing selection, or selection on standing selection among others). From the genetic variation point of view, selection can influence in several different ways to increase, decrease, or maintain diversity *[Jobling 2014].*

Demographic factors

Nevertheless, according to the neutral hypothesis of evolution, most of the mutations that occur in the genome do not have a functional impact in the phenotype. The fate of these mutations in the population depends on genetic drift, which refers to the stochastic process of sampling due to the finite number of chromosomes that replicate at each generation. Suppose that a pool of gametes contains the alleles “A” and “a” at frequencies “p” and “q” with p+q=1. Then if 2N gametes are drawn at random to produce the zygotes of the next generation, the probability that the sample contains exactly “j” alleles of type A is as follows *(Equation 1.1) [Hartl 2007]:*

$$\Pr\{j \text{ alleles of type } A\} = \binom{2N}{j} p^j q^{2N-j} = \frac{(2N)!}{j!(2N-j)!} p^j q^{2N-j}$$

The smaller the population size that reproduces at each generation, the higher the random sampling process at each generation and higher the fixation/erasing rate of mutations.

In this context, the effective population size is defined as the number of random mating individuals in an ideal population compared to the real population *[Jobling 2014].* There are two mathematical ways of defining effective population sizes: one is based on the sampling variance of allele frequencies (that is, how an allele’s frequency might vary from one generation to the next), and the other utilizes the concept of inbreeding (that is, the probability that the two alleles within an individual are identical by descent from a common ancestor). Both of these properties of a finite population depend on the mating size of that population. There also can be non genetic definitions, such as the number of breeding individuals inferred from demographic studies.

To illustrate the effect of these factors on the definition of the effective population size of one species, we can take a look at cattle in North America: there are about 100,000,000 female cattle in North America fertilized on average by four males through artificial insemination. Therefore, having four bulls that are inseminating 100,000,000 cows, genetically speaking the effective population size is just about 16 *[Stearns 2010].*

1.2 Theoretical models of spatially structured populations.

Natural populations have complex geographies and histories and the problem of local differentiation of gene frequencies in a structured population has historically required the use of basic models allowing the development of the explanatory mathematical theory. Despite the simplicity of these ideal prototypes, the design of very basic theoretical models of spatially structured populations has provided useful intuition about the behavior of more complex models.

A major development in terms of modelling strategies of spatially structured populations was the use in the 1960's by Kimura and others of island and stepping stone models *[Kimura 1964, Slatkin 1975]*

The term "island model" refers to a model which considers a population to be split into a finite or infinite number of discrete islands or demes. Individuals can migrate from any deme to any other deme uniformly, there is no sense of distance and all islands are equally far apart from each other (Fig 1.1).

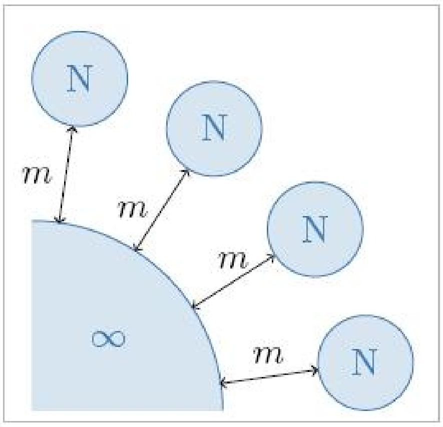


Figure 1.1: Each island has a finite diploid population of size N, each of which exchanges a proportion "m" of its population each generation with a mainland containing an infinite population. We define M = 2Nm for the total number of migrants exchanged per generation.

In this context, a useful and related statistic to measure the differentiation between subpopulations is Wright's fixation index:

$$\boxed{F_{ST} = \frac{\sigma_S^2}{\sigma_T^2} = \frac{\sigma_S^2}{\bar{p}(1-\bar{p})}}$$

Where "p" is the average frequency of an allele in the total population, $\sigma^2{}_S$ is the variance in the frequency of the allele between different subpopulations, weighted by the sizes of the subpopulations, and $\sigma^2{}_T$ is the variance of the allelic state in the total population.

Although island models have the advantage of being manageable, they do not correspond to reality. Most populations will exhibit isolation by distance and the assumption that an individual from any part of the range migrates uniformly is likely to be false. Individuals are more likely to be closely genetically related to individuals who are also spatially close to them (keeping alive the first law of geography: *"Everything is related to everything else, but near things are more related than distant things"*). The "stepping stone" model *[Kimura 1964]* is the natural extension of the island model by including the concept of isolation by distance in the model concept. In the stepping stone model, demes are arranged in a regular pattern, migration can only occur between the nearby neighbor surrounding demes and the degree of differentiation between subpopulations depends on the number of populations.

Formally, if we have "d" demes arranged in a K x K grid *[Cox 2002],* genetic differentiation among demes is:

$$F_{ST} \approx \frac{1}{1 + \frac{4Nm\pi}{\log(d)}}$$

While the one dimensional model can represent a population of organisms living along a river, the two dimensional model can represent a population on a plane and cover the most important cases in nature including different migration rates in the longitude X and latitude Y axis directions. The three dimensional model can represent a population in an oceanic habitat with migration rates in the three axes or can also represent a population of organisms living on a plane, but including a third dimension such as the social rank in which migration is restricted to the neighboring classes *[Kimura 1964]*.

Isolation by distance theory explains the accumulation of local genetic differences under the main driver "the more geographical distance, the more genetic differentiation in the pairwise measures". However, we need to take into account that genetic differentiation can increase at different rates in different geographic directions, and this should affect the localization of geographic origin from genome-wide SNP data. Keeping in mind that the concept of anisotropy is defined as the property of being directionally dependent, we can rely on spatial analysis of genetic data since can additionally provide the orientation at which the accumulation of genetic differentiation is the greatest *[Jay 2013]*. In fact, main land-masses do not show the same orientation and this creates an effect of anisotropy in the spatial distribution of the genetic variation.

1.3 The coalescent approach for modeling the neutral genetic variation.

Wright-Fisher model.

The Wright-Fisher model is one of the most simplest models for modeling the observed demographic variation of a population. The Wright-Fisher model makes three idealized assumptions: (i) Generations are taken to be discrete (ii) The population size is taken to be fixed, so that alleles compete only against other alleles and not against an external environment. (iii) Random mating is assumed. None of these assumptions are present in any real population. Nevertheless, Wright-Fisher has proved to be a useful intuitive guide in real cases, and also the mathematical foundation on which more complicated population models can be developed.

Coalescent theory

The term coalescence refers to the process in which, looking backward in time, the genealogies of two alleles at present merge into a shared common ancestor in the past. In a sample of “k” alleles, for example, the first coalescence (looking backward in time) merges the “k” contemporary genealogies into (k - 1) ancestral genealogies, and the second coalescence merges these into (k - 2) genealogies, and so forth, until there remains a single common ancestor for the whole sample of alleles at present. The idea of coalescent analysis is to consider the ancestral history of genes in a sample by developing a model for the time intervals between each coalescence.

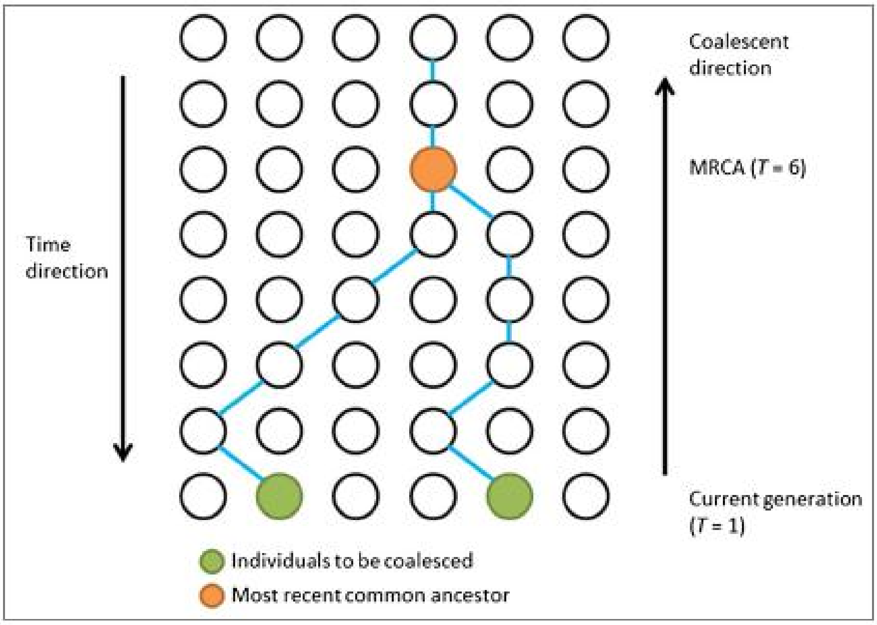

Coalescent direction
MRCA ($T = 6$)
Time direction
Current generation ($T = 1$)
Individuals to be coalesced
Most recent common ancestor

Figure 1.1: In a Wright-Fisher model, two haploid individuals (green) at the present generation coalesced at the sixth generation backward in time to the most recent common ancestral - MRCA (orange). Extracted from [Critical assessment of coalescent simulators in modeling recombination hotspots in genomic sequences, Tao Yang 2014]

The coalescent theory was initially derived as an approximation for explaining the patterns of variation observed at one locus taking advantage of the neutral Wright-Fisher model: given an effective population size and a sample of alleles, we can estimate each coalescence tree in probability following the *(Equation 1.1)* backward in time. Once the tree is constructed, we can estimate the probability of the observed genetic diversity in our samples by adding mutations to this tree. These mutations would follow a Poisson process where the scaling factor of a branch would be determined by the number of generations since the last coalescence event. Normally, an infinite sites model is assumed, which means no recurrent mutations occur. Each recombination event breaks the sequence into several segments, and each segment is modeled by a genealogy tree. Simulation of recombination hotspots is realized by changing the rates where these recombination events occur *[Yang and Deng 2014].*

This approximation works well when sample sizes are small relative to the effective population size.

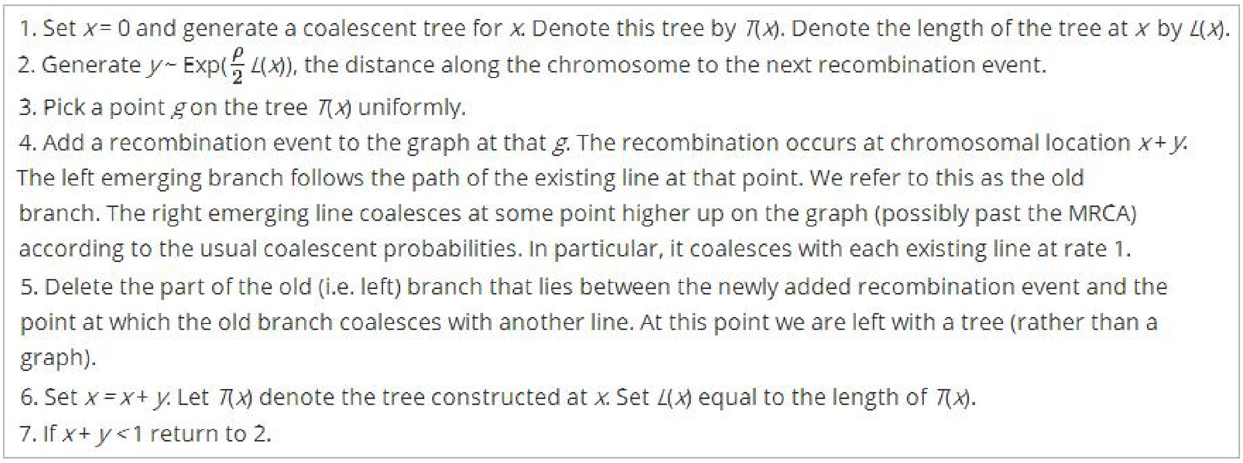
2. Applying the coalescent theory to simulate sequences.

Backward coalescent simulations are the standard method of generate population samples under various demographic models. They are widely used as powerful tools in the field of population genetics and are the keystone in estimating parameters for different population histories, to infer phylogenetic trees, testing against the presence of selective sweeps and for providing an evaluation framework at association studies among others. Based on the neutral Wright-Fisher model, the backward coalescent simulation process starts from a sample of DNA sequences and then integrates all coalescent and recombination events simulating the entire ancestral origin. From the computational point of view, the process imposes considerable computational requirements. Since the year 2002, an abundant number of coalescent simulators have been developed and, among them, Fastsimcoal2 is one of the more scalable and flexible.

Fastsimcoal2 is a program to simulate the neutral genomic molecular diversity in current or ancient samples derived from a population given a demographic model. Fastsimcoal2 generates replicates of random outcome of molecular diversity under a user-defined evolutionary scenario. These scenarios can be very complex from the evolutionary point of view, including an arbitrary migration matrix between samples, historical events allowing for population resize, population fusion and fission, admixture events, changes in migration matrix, or changes in population growth rates *[Excoffier 2011].*

Fastsimcoal2 is fitted with a fast Sequential Markovian Coalescent (SMC) model for recombining DNA sequences, in particular the SMC' version of SMC *[Marjoram 2006].* Under SMC, a tree is generated on the left end of the sequence under study, and computes the position of a recombination event on the right-hand side assuming an exponential distribution of recombination positions along the sequence. A recombination event is then implemented at random along the current tree, and the detached recombining lineage is then free to coalesce with the other remaining lineages, leading to a new tree with a potentially different topology and most recent common ancestor (MRCA). This procedure is continued until one reaches the end of the sequence to be generated. By the implementation of SMC' algorithm, for each tree, all migration events having occurred in addition to all coalescent events are recorded. These events are then replayed to generate the next tree, such that the detached recombinant lineage can migrate in any deme and potentially coalesce with lineages from the left tree that were present there at the same time *[Excoffier 2011].*

Due to the clarity how SMC' algorithm is described by the authors *[Marjoram 2006].* we transcribe it here in full:



1. Set $x = 0$ and generate a coalescent tree for x. Denote this tree by $T(x)$. Denote the length of the tree at x by $L(x)$.
2. Generate $y \sim \text{Exp}(\frac{\rho}{2} L(x))$, the distance along the chromosome to the next recombination event.
3. Pick a point g on the tree $T(x)$ uniformly.
4. Add a recombination event to the graph at that g. The recombination occurs at chromosomal location $x + y$. The left emerging branch follows the path of the existing line at that point. We refer to this as the old branch. The right emerging line coalesces at some point higher up on the graph (possibly past the MRCA) according to the usual coalescent probabilities. In particular, it coalesces with each existing line at rate 1.
5. Delete the part of the old (i.e. left) branch that lies between the newly added recombination event and the point at which the old branch coalesces with another line. At this point we are left with a tree (rather than a graph).
6. Set $x = x + y$. Let $T(x)$ denote the tree constructed at x. Set $L(x)$ equal to the length of $T(x)$.
7. If $x + y < 1$ return to 2.

Figure 2.1: The seven steps of the SMC' Algorithm. Extracted from [Fast coalescent simulation, Marjoram and Wall].

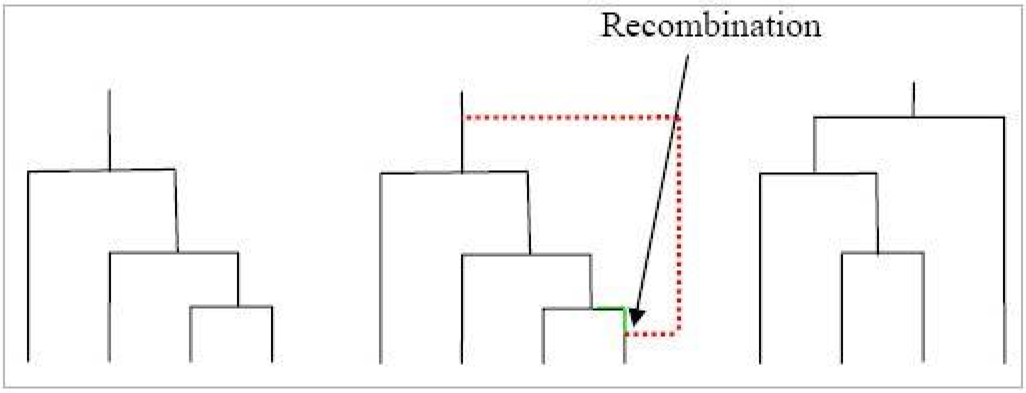


Figure 2.2: how the SMC' algorithm forms the next tree along the chromosome, moving from left-to-right, given the state of the current tree. Extracted from [Fast coalescent simulation, Marjoram and Wall].

3. Methods for detecting global population substructure.

3.1 The analysis of population structure.

Population structure is produced when the population or metapopulation is subdivided in sub-populations, local populations, or demes whose individuals randomly reproduce at a higher rate within each deme than between demes.

Identifying the sub-populations that comprise a species and the gene flow connections between different demes is an active research field within population genetics and has important consequences for properly interpreting the genetic diversity, for instance identifying genetic/geographic barriers or discontinuities. Ultimately, we can not assume that genetic variation over the whole range of a species is simply the same as extrapolating what happens in single populations. In other words: population structure matters.

In many real populations, population substructure may be cryptic and/or show continuous spatial patterns. However, even in effectively spatially continuous environments, different geographic areas can differ in gene frequencies, because the whole metapopulation is not panmictic. For instance, among humans, there are regions showing some quite major language differences, suggesting substructure, but you would be hard put to find an exact boundary where there is a changeover. Such populations are structured, but continuously, in space.

The analysis of population structure based on genetic ancestry has experienced an increasing progress during the last decade. “Genetic ancestry estimation” is a broad term which is concerned with a number of different population genetics problems *[Liu 2013]* :

- defining the number of subpopulations in a sample
- assigning individuals to subpopulations
- defining the number of ancestral populations in admixed populations
- assigning ancestral population proportions to admixed individuals
- identifying the genetic ancestry of distinct chromosomal segments within an individual

This information can be further used to inform us about the evolutionary relationships and migration history of natural populations. In the case of humans, where both the sampling location of an organism or self-reported ancestry can be uninformative for the true ancestry of the individual, the use of genetic markers can facilitate accurate and reliable ancestry inference by exploiting allele frequency differences across population groups.

3.2 Local and global ancestry estimation.

Taking into account that the chromosomes of an individual with admixed ancestry represent a mosaic of chromosomal blocks from the ancestral populations (see Figure 3.1), there are currently two different paradigms underlying ancestry inference: global ancestry estimation and local ancestry estimation:

Local Ancestry Estimation. Local ancestry is defined as the genetic ancestry of an individual at a particular chromosomal location, where an individual can have 0, 1 or 2 copies of an allele derived from each ancestral population. Local estimates are concerned with identifying the ancestral origin of distinct chromosomal segments within an individual genome and, henceforth, analysing each chromosome in an individual's genome as a mosaic of segments that originate from different ancestral populations *[Padhukasahasram 2014]*

Global Ancestry Estimation: Global ancestry is based on estimating the proportion of ancestry contributed by different populations averaged across the entire genome of an individual. Despite this estimation can be obtained by averaging the ancestry tracts obtained from local ancestry methods, there is a large number of algorithms that tackle genome ancestry problem as a whole.

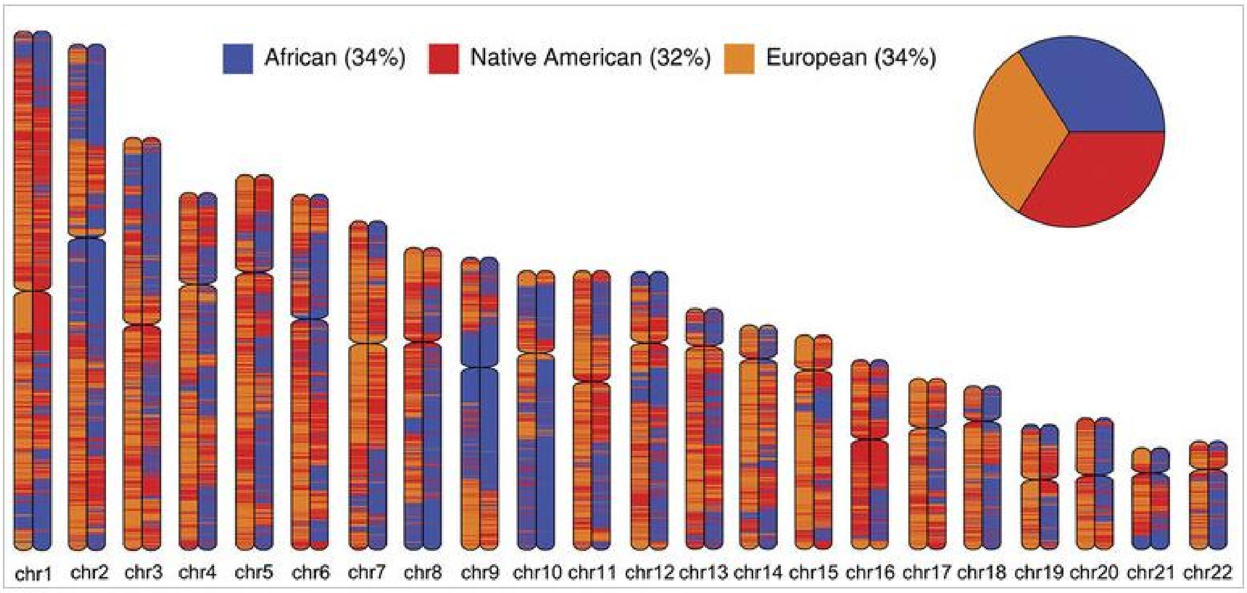


Figure 3.1: Example of local genetic ancestry. Chromosome paintings showing the genomic distributions of loci with African, Asian (Native American) and European ancestry, along with their genome-wide ancestry proportions for one particular Colombian individual. Extracted from [Ancestry, admixture and fitness in Colombian genomes, Lavanya Rishishwar]

Despite no single method or software can optimally solve all of these problems *[Padhukasahasram 2014]* recent advances in genomic technologies as well as computing resources have made it possible to accurately infer overall ancestry as well as ancestry at a fine scale across an individual's genome.

Apart from global and local ancestry approaches, under the large topic of global ancestry estimation the algorithms for estimating genetic ancestry can also be divided into methods that rely on multivariate statistical methods (like PCA and cluster analysis) versus methods that make use of explicit genetic models. However, this distinction does not imply that there aren't important similarities between algorithmic and model-based methods *[Liu 2013].*

3.3 "Algorithmic" based methods.

Algorithmic approaches use techniques from multivariate analysis, mainly cluster analysis and principal component analysis, to discover structure within the data not making any assumption about the underlying genetic model of the data. The proposed output of some of these methods (coordinates) can be interpreted in demographic terms.

Algorithm free methods are exemplified by MDS-PLINK *[Sham 2007]* , SMARTPCA-Eigensoft *[Patterson 2006]* or sNMF among others (for an extensive overview of most widely used methods in population genetics for detecting individual genetic ancestry, see *Detecting individual ancestry in the human genome, Wollstein and Lao]*)

The Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is an algorithm that iteratively searches for orthogonal axes, described as linear combinations of multivariate observations, along which projected objects show the highest variance, and then returns the positions of objects along those axes (the principal components). For many data sets, the relative position of these objects (e.g., individuals) along the first few PCs provides a reasonable approximation of the covariance pattern among individuals in the larger data set. As a result, the first few PC values are often used to explore the structure of variation in the sample.

A principal component analysis makes sense if there are high correlations between variables, as this is indicative that there is redundant information and therefore fewer factors explain as much of the variability as the total set of variables. This is the case of genetic variants: because demographic processes affect the whole genome, it is expected that a large number of variants will correlate due to their shared history (see Origin and maintenance of the genetic variation in the human genome).

The selection of factors is performed such that the first factor collects the largest amount of the original variability between observations; the second factor collects the maximum possible variability not collected by the first, and so on (see Figure 3.2). From all these factors we can select those that collect the percentage of variability that is considered sufficient for the analysis. Once selected the main components, they are represented in a matrix where each element of this factor represents the coefficients of the variables (correlations between variables and principal components). The matrix will have as many columns as main components and as many rows as variables. *[Peña 2002]*

The PCA problem can be approached from a geometric point of view if we consider the point cloud from our dataset and we see that the points are located following an ellipse, then we can describe its orientation giving the direction of the major axis of the ellipse and the position of the point by its projection on this direction. It can be shown that this axis is the line that minimizes the orthogonal distances. In several dimensions, prior concept can be applied to ellipsoids and the best approach to data is provided by the major axis of the ellipsoid. Considering the ellipsoid axes as new variables of moving from original variables correlated to orthogonal variables.

Formally, let X be a $m \times n$ matrix with observations, each of dimensionality m, such that each variable has zero mean, $\Sigma Xij = 0 \; \forall i$. Then the principal components (PCs) Y are given by the transformation $Y = W^{T} X$ where W is an orthogonal $m \times m$ matrix chosen as follows: let w_{i} be the i th column of W, then w_{1} satisfies $w_{1} = argmax \,||w1||_{=1}\{||w^{T}_{1} X||^{2}\}$ and for $1 < i \leq m$, w_{i} satisfies $w_{i} = argmax \,||wi||_{=1}\{||w^{T}_{i}[I - \Sigma j_{j=1}^{i-1} w_{j} w^{T}_{j}]X||^{2}\}$. Equivalently, the principal components are the ordered eigenvectors of the sample covariance matrix.

PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

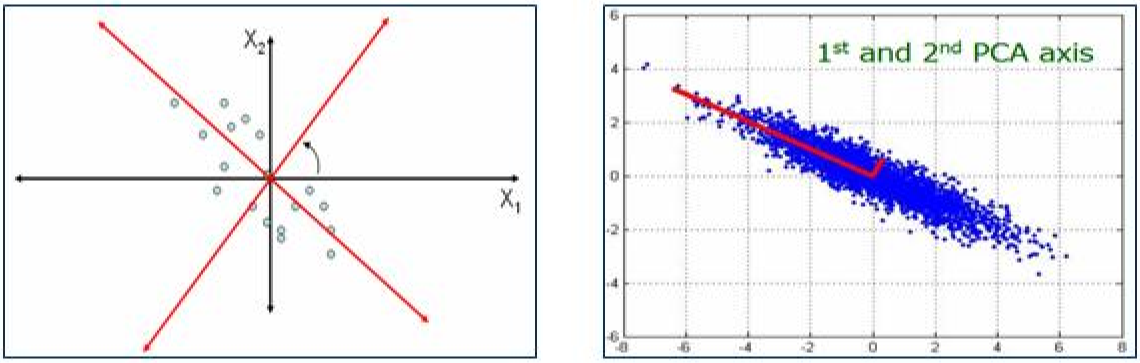


Figure 3.2: Illustration of PCA principles in a two dimensional (x1 and X2) example. Each dot represents an observation in these two dimensions. Left panel: Transforming to new coordinate system (red axis) with major variance on the new first coordinate Right panel: Observed variables are projected onto their two principal components resulting in a set of orthogonal predictors being the best approach provided by the major axis of the ellipsoid.

Mathematically, the transformation is defined by a set of p-dimensional vectors of weights or loadings $w(k) = (w_{1}, \ldots., w_{p})(k)$ that map each row vector $X(i)$ of X to a new vector of principal component scores $t(i) = (t_{1}, \ldots., t_{p})(i)$ given by $tk(i) = X(i) \cdot w(k)$ in such a way that the individual variables of t

considered over the data set successively inherit the maximum possible variance from X, with each loading vector w constrained to be a unit vector.

History of PCA in population genetics: Coalescent interpretation, difficulties and new uses.

The application of PCA to genetic data was leaded by Cavalli-Sforza in the mid-1960s. From a wide range of populations and using relatively small number of classical markers (mostly blood groups and related biological markers) available at that time, Cavalli-Sforza investigated the structure and relationships between different human groups.

In the 70's PCA was commonly used to visualise genetic data using low size datasets; however, the method was mostly abandoned in the 80's and 90's for these purposes due to problems related to interpretation of PCA *[Sokal 2012]* and data used not meeting assumptions of PCA. In 2006 *[Price 2006]*, with the development of high-density SNP genotyping assays which has made possible to characterize patterns of genetic variation within and among human populations, the method was reintroduced in the population genomics community, providing unprecedented opportunities to understand the evolutionary history and migration patterns of humans. Such datasets with an order of magnitude 10^6 absolutely require a dimensionality reduction technique to be summarised and visualised, and PCA is usually the most convenient. *[Jianzhong Ma 2012]* PCA is widely used to quantify patterns of population structure and the Eigenstrat method, as implemented in the program SmartPCA, is now routinely used to detect and correct for population stratification in genome-wide association studies *[Patterson 2006].*

In conventional PCA, in which the markers are treated as features, sampled individuals are projected into a subspace spanned by the top principal components (PCs). Because the top PCs reflect variations due to population structure in the sample, individuals from the same population are found to form a cluster in this subspace. Therefore, the pattern of the top PCA is used to infer population relationships or within-population structures that can be understood intuitively.However, the biological interpretation of principal components from genetic data is non-obvious. Cavalli-Sforza and colleagues interpreted variation in principal components in evolutionary terms, so that a PCA component in the geographic space was indicative of :

- an admixture event

- a selective gradient

- a migration event

- a range expansion

Nevertheless, as was pointed out by Novembre and Stephens *[Novembre 2006]*, these PCA clines and other more complex regular patterns also appear naturally in the first few principal components of variation when spatially structured populations are at both equilibrium and nonequilibrium models. According to Novembre and Stephens [Novembre 2006], when analyzing spatial data:

- PCA produces highly structured results, in particular sinusoidal functions of increasing frequency.

- PCA results depend on the details of a particular dataset:

  > population structure

  > distribution of sampling locations

  > amounts of data

- These features limit the utility of PCA for drawing inferences about underlying processes and interpreting gradient patterns in PC maps as signatures of historical migration event, because such patterns arise generally under a simple condition:

  > genetic similarity decays with distance and this condition would be expected to be satisfied under a wide range of demographic scenarios, including both equilibrium isolation-by distance models and nonequilibrium models involving population expansions.

- Because Cavalli-Sforza et al. used spatial interpolation to estimate allele frequencies, their data could satisfy this condition even if the condition were absent in the underlying allelic frequencies [Novembre 2006].

This is a problem to be further discussed. For example, the first two principal components of European genotypes almost perfectly recreate geographic North-South and East-West axes [Lao 2008] but it is not clear whether this is a result of range expansion in both these directions, constant population structure with migration, or the most likely option, a combination of both.

McVean (2009) provided a unifying framework for understanding what PCA actually represents in a genomic context, by showing that the principal components are simply a function of the expected coalescence times between lineages (see The coalescent approach for modeling the neutral genetic variation). Thus, models which lead to the same expected coalescence times provide the same PCA output. Furthermore, an additional major problem when using PCA to analyse genetic data is that it is very sensitive to both ascertainment of markers and sampling scheme. Choosing different individuals can lead to very different conclusions, and according to McVean [McVean 2009] the main drivers for this situation are:

- PCA projections can be strongly influenced by uneven sampling from a series of populations. If all populations are equally divergent from each other, those for which there are fewer samples will have larger values because relatively more pairwise comparisons are between populations.

- Even if the results were not influenced by the relative sample size its eigenvectors will be, simply because relative sample size will influence the structure of the genetic variance in the sample.

- The influence of uneven sample size can bias the projection of samples on the first few PCs in unexpected ways, for example, where there is spatial structure to genetic variation.

- there are many different processes that one might want to consider as explanations for patterns of structure in empirical data and efficient inference, even under simple models can be difficult.

- Different processes can lead to similar patterns of structure. For example, equilibrium models of restricted migration can give similar patterns of differentiation to non-equilibrium models of population splitting events.

- Any species is likely to have experienced many different demographic events and processes in its history and their superposition leads to complex patterns of genetic variability.

- Such models are often highly simplistic and restricted to a subset of possible explanations.

Multidimensional scaling (MDS)

The multidimensional scaling techniques (MDS) are a generalization of the idea of principal components when, instead of having a matrix of observations by variables such as principal component, there is a square nxn matrix "D" of distances or dissimilarities between the "n" elements of a dataset. These distances may have been obtained from certain variables or may be the result of a direct estimate. The objective is to represent this matrix by using a set of orthogonal variables (y1,....yp) where p<n so the Euclidean distances between the coordinates of the elements on these variables are equal (or as close as possible) to the distance or dissimilarity of the original matrix. That is, from the matrix "D" it is obtained a "X" matrix n×p, which can be interpreted as the matrix of "p" variables in the "n" individuals, and where the Euclidean distance between the elements approximately reproduces the initial distance matrix "D". In general it is not possible to find "p" variables that reproduce exactly the initial distances, however it is common to find variables that reproduce approximately the initial distances. On the other hand, if the distance matrix was generated by calculating the Euclidean distances between observations defined by certain variables, we can recover the main components of these variables *[Peña 2002]*.

The multidimensional scaling shares its main goal with principal components in order to synthesize the individual relationships and the interpretation of the data. If there are many elements, the matrix of similarities will be very large and the representation by a few variables elements will allow us to understand its structure: what elements have similar properties or groups appearing between elements.

<u>Multidimensional Scaling (MDS) on Identity By State (IBS) pairwise matrix.</u>

Population substructure modeling of a sample individuals can be done by computing the Identity By State (IBS) distance between each pair of individuals, subsequently building a matrix representing the relatedness of individuals and then performing multidimensional scaling (MDS) using such matrix. IBS examines pairs of SNPs between two individuals and puts them into one of three categories (see Figure 3.3):

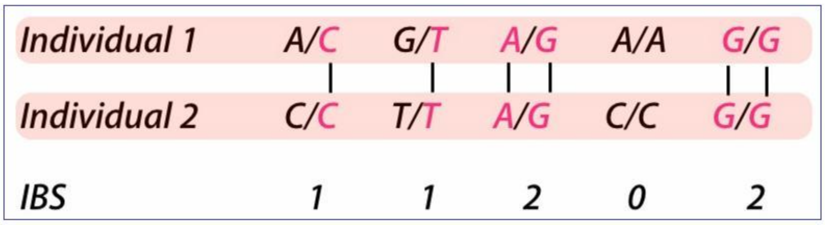
1. Identical: Both individuals have the same genotype call (AA and AA; BB and BB; AB and AB).

2. One-Allele Shared: Only one call is shared between both individuals (AA and AB; AB and BB).

3. No alleles shared: No alleles are the same (AA and BB).

For individual SNPs, this type of analysis really does not provide any extra information. The real advantage is gained when high-density SNP information is taken for the whole genome.

Conceptually, having N loci, what we are really doing is plotting the individuals as points in a N-dimensional space where each individual's distance from another along each axis is either 0, 1, or 2 (IBS-0 is a distance of 2, IBS-1 is a distance of 1, IBS-2 is a distance of 0) and then computing the distance between each pair of points along each axis. Those distances will only plot properly in a N-dimensional space, but, with higher dimensions being difficult to visualize, we can use MDS to plot an approximation of the distances in 2D and iteratively try to find a positioning of the points in two dimensions that minimizes conflict between their true distance and their distance as plotted.



Individual 1	*A/C*	*G/T*	*A/G*	*A/A*	*G/G*
Individual 2	*C/C*	*T/T*	*A/G*	*C/C*	*G/G*
IBS	*1*	*1*	*2*	*0*	*2*

Figure 3.3: IBS classification example showing the three possible categories by examining pairs of SNP's between two individuals.

IBS is suitable for population outlier detection, is robust to high linkage disequilibrium (LD) among SNPs, and can be rapidly calculated *[Gao 2009]*. Furthermore, one of the advantages of this distance method is that there is no need to explicitly specify the allele frequencies. Therefore, population allele frequencies do not have to be approximated by sample allele frequencies. The allele frequencies and coancestry information are embedded in the pairwise distance matrix over a large number of random SNP loci. *[Gao 2009]*. Another advantage of the distance method is that it is easy to calculate with no decrease in accuracy and is also suitable for population outlier detection.

In summary, the IBS method combined with SNP markers has considerable power in population stratification analysis and it is not necessary to estimate allele frequencies to separate individuals with different ethnic backgrounds. The correlation/coancestry among individuals within subpopulations, which can be captured by the IBS, contributes to the classification. Diploid individuals from different subpopulations can thus be separated from half-matrix of pairwise distances.

SNMF

SNMF program *[Frichot 2014]* applies an algorithm for inferring ancestry proportions founded on the linear algebra principle of non-negative matrix factorization (see Figure 3.4).

Non-negative matrix factorization (NMF) is a group of algorithms in multivariate analysis where a matrix V is factorized into two matrices W and H, with the property that all three matrices have no negative elements. Apart from the obvious fact that non negative values make the resulting matrices easier to inspect, in NMF applications such as nuclear imaging, processing of audio spectrograms or, as in our case, detecting individual ancestry in the human genome, non-negativity is inherent to the data being considered. Since the problem is not exactly solvable in general, it is commonly approximated numerically. Non-negative matrix factorization is distinguished from the other methods by its use of non-negativity constraints which lead to a parts-based representation because they allow only additive, non subtractive, combinations *[Lee 1999].*

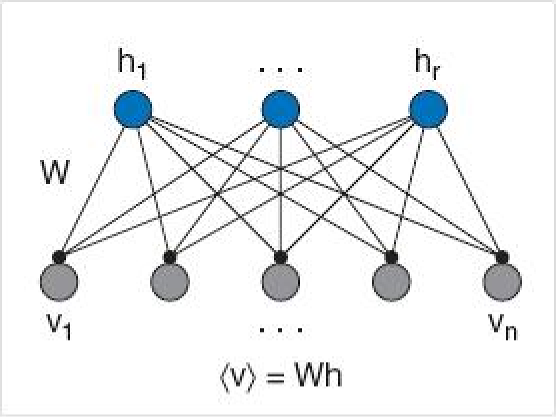


Figure 3.4: Probabilistic hidden variables model: the visible variables "v" in the bottom layer of nodes are generated from the hidden variables "h" in the top from a probability distribution with mean $\Sigma a(Wia \cdot ha)$. The influence of h_a on v_i is represented by a connection with strength W_{ia} . [Daniel D. Lee and H. Sebastian Seung (1999). "Learning the parts of objects by non-negative matrix factorization". Nature. 401 (6755): 788–791]

The algebraic basis of NMF is as follows: let matrix V be the product of the matrices W and H,

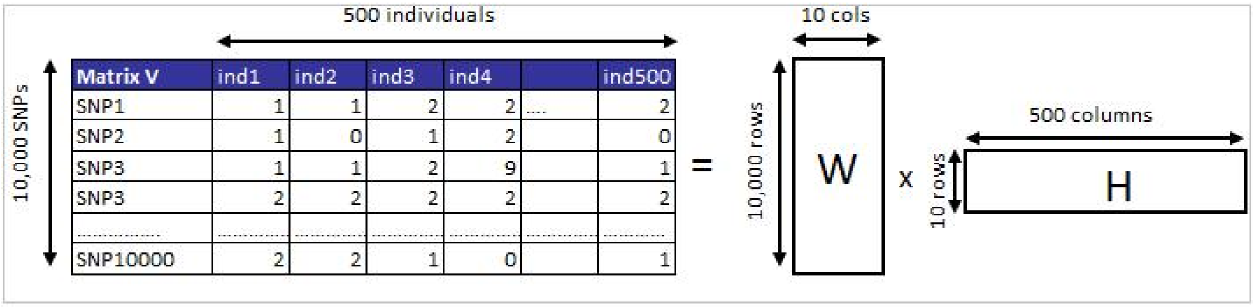
$$V = WH$$

and when multiplying these matrices, the dimensions of the factor matrices (W and H) may be significantly lower than those of the product matrix (V) and it is this property that forms the basis of

NMF: generating factors with significantly reduced dimensions compared to the original matrix. For example, if V is an $m \times n$ matrix, W is an $m \times p$ matrix, and H is a $p \times n$ matrix then p can be significantly less than both m and n.

For illustrative purposes, a simplified example for detecting individual ancestry in the human genome by using NMF algorithm would follow next steps:

S1) Let the input matrix (the matrix to be factored) be "V" with 10,000 rows and 500 columns where SNPs are in rows and individuals are in columns *(Figure 3.5)*. Each row contains 1 character per individual: 0 means zero copies of the reference allele. 1 means one copy of the reference allele. 2 means two copies of the reference allele. 9 means missing data:



Matrix V	ind1	ind2	ind3	ind4		ind500
SNP1	1	1	2	2	....	2
SNP2	1	0	1	2		0
SNP3	1	1	2	9		1
SNP3	2	2	2	2		2
....	....	....	....	....	....	....
SNP10000	2	2	1	0		1

Figure 3.5: Illustrative example of sNMF factorization. The input matrix to be factored, a 10,000 SNPs genotyped at 500 individuals from the left of the equation (SNP's in rows, individuals in columns) is decomposed in two non-negative features matrix W and coefficients matrix H on the right.

S2) Assume we ask the algorithm to find -lets say- 10 features in order to generate a features matrix "W" with 10,000 rows and 10 columns and a coefficients matrix H with 10 rows and 500 columns. The product of W and H is a matrix with 10000 rows and 500 columns, the same shape as the input matrix V and, if the factorization worked, it is a reasonable approximation to the input matrix V.

S3) From the treatment of matrix multiplication above it follows that each column in the product matrix WH is a linear combination of the 10 column vectors in the features matrix W with coefficients supplied by the coefficients matrix H.

This last point is the basis of NMF because we can consider each original individual in our example as being built from a small set of hidden features. NMF generates these features.

S4) We can interpret each feature (column vector) in the features matrix W as an individual archetype comprising a set of SNPs where each SNP cell value defines the SNP's rank in the feature: The higher a SNP's cell value the higher the SNP's rank in the feature. A column in the coefficients matrix H represents an original individual with a cell value defining the individual's rank for a feature. This follows because each row in H represents a feature. We can now reconstruct an individual (column vector) from our input matrix by a linear combination of our features (column vectors in W) where each feature is weighted by the feature cell value from the individual's column in H.

In the context of the above mathematical framework, sNMF software models the probability of the observed genotypes p_{il} in individual "i" at locus "l" as a fraction q_{ik} of K ancestral genotype probability g_{kl} *[Lao and Wollstein 2015]*:

$$p_{il}(j) = \sum_{k=1}^{K} q_{ik} g_{kl}(j)$$

As exemplified above, j=0,1,2 denotes the number of alleles. The corresponding matrix representation is P=QG, where the unknown Q and G can be estimated by nonlinear matrix factorization minimizing two least square criteria:

$$Ls_1 = |X - QG| \text{and} \, Ls_2 = \left|(G^T; \sqrt{\alpha}\, 1_K)Q^T - (X^T; 0_n)\right|$$

A loop is then executed applying both criteria until convergence is reached. Starting from random matrices as initial condition, the algorithm finally obtain estimates about Q from Ls_1 and G from Ls_2. *[Wollstein and Lao 2015]*.

3.4 “Model” based methods.

Model-based approaches estimate individual ancestry proportions and ancestral populations as the parameters of a statistical model. Model based algorithms philosophy are exemplified by STRUCTURE *[Pritchard 2000]*, FRAPPE *[Tang 2005]* or ADMIXTURE *[Alexander 2009].*

In 2000 the seminal paper of Pritchard et al *[Pritchard 2000]* introduced STRUCTURE, a new method for identifying individual global ancestry. The method was based on estimating ancestry proportions from a putative number of ancestral populations that produced currently observed data by assuming very basic population genetic assumptions and implementing a Bayesian framework to recover the ancestry proportions of each individual as well as the ancestral allelic proportions in the ancestral populations.

From a conceptual point of view, this method revolutionized the analysis and interpretation of human genomic data. First of all, it showed that ancestry proportions could be recovered at individual level, rather than at a population level. Second, in 2002, Rosenberg et al *[Rosenberg 2002]* showed that continental like groups of humans could be identified by means of using this method through STRUCTURE software, without using prior information about the origins of individuals. The authors identified six main genetic clusters, five of which corresponded to major geographic regions, and sub-clusters that corresponded to individual populations. Finally, since its publication, this -and maximum likelihood based approaches- became the gold standard for identifying population substructure when analyzing genetic data.

From a methodological point of view, this new method introduced by Pritchard (STRUCTURE) is based on the assumption of basic demographic assumptions and estimating ancestry coefficients as the parameters of a statistical model. *[Pritchard 2000].*

Formally and in brief, STRUCTURE algorithm assumes:

- Each cluster or population is modeled by a specific set of allele frequencies.

- HWE within populations.

- Complete linkage equilibrium between loci within populations.

- Each allele at each genotype is an independent draw from the appropriate frequency distribution Pr(X|Z,P) where X denote the genotypes of the sampled individuals, Z denote the unknown populations of origin of individuals and P denote the unknown allele frequencies in all populations. See next point for a description of the individual and population likelihood.

ADMIXTURE

The approach for estimating ancestry proportions of ADMIXTURE software is similar to STRUCTURE methodology since both programs model the probability of the observed genotypes using ancestry proportions and population allele frequencies. Like STRUCTURE, ADMIXTURE simultaneously estimates population allele frequencies along with ancestry proportions *[Alexander 2009]:*

in the likelihood model, individuals are formed by the random union of gametes producing the binomial proportions:

$$\begin{aligned}
\Pr(1/1 \text{ for } i \text{ at SNP } j) &= \left[\sum_k q_{ik} f_{kj}\right]^2 \\
\Pr(1/2 \text{ for } i \text{ at SNP } j) &= 2\left[\sum_k q_{ik} f_{kj}\right]\left[\sum_k q_{ik}(1-f_{kj})\right] \\
\Pr(2/2 \text{ for } i \text{ at SNP } j) &= \left[\sum_k q_{ik}(1-f_{kj})\right]^2
\end{aligned}$$

Being g_{ij} the observed number of copies of allele "1" at marker "j" of individual "i" which equals 2, 1 or 0 if "i" has genotype 1/1, 1/2 or 2/2 at marker "j", since individuals are considered independent, the log-likelihood of the entire sample is:

$$L(Q,F) = \sum_i \sum_j \left\{ g_{ij} \ln\left[\sum_k q_{ik} f_{kj}\right] + (2-g_{ij}) \ln\left[\sum_k q_{ik}(1-f_{kj})\right]\right\}$$

In this expression, Q = (q_{ik}) represents the matrix of ancestry coefficients for all individuals, and F = (f_{kj}) represents a matrix of allele frequencies for all loci.

The main difference between ADMIXTURE and STRUCTURE relies on maximizing the likelihood rather than on sampling the posterior by MCMC as done by STRUCTURE. Since high-dimensional optimization is much faster than high-dimensional MCMC, ADMIXTURE maximum likelihood approach can accommodate many more markers. The parameters of the ADMIXTURE model must satisfy linear constraints and bounds and is settled on a block relaxation algorithm that alternates between updating the ancestry coefficient matrix Q and the population allele frequency matrix F. Each update of Q itself involves sequential quadratic programming, a generalization of Newton's method suitable for constrained optimization.

Since model-based methods explore the space of possible solutions starting from an stochastic initial point, it is generally suggested to run the algorithm several times at different initial starting points for each proposed K and to check for the optimal resulting scenario. Different strategies have been proposed for combining the results from different runs: merging all the solutions and then computing a consensus ancestry value or just to take the run that provides the best value of model performance *[Wollstein and Lao 2015]*.

Identification of the optimal number of ancestral populations

The identification of the number of ancestral populations contributing to current genetic variation is of interest for several population genomic fields such as association mapping, molecular ecology or human evolution studies among others. Many algorithms have been developed for employing population genetic data to estimate the individual ancestral proportions out of a predefined set of K ancestral populations. Typically, a matrix structure is used to represent the individual ancestry proportions over all the samples, where each individual is given a coefficient or fraction for each cluster, all adding to 1. This fraction can have multiple interpretations. In one hand, it can be interpreted as the probability of being a member of the ancestral population. On the other hand, it can indicate the fraction of the genome with membership in the ancestral population. The number of ancestral components is usually predefined by the user for some methods, and a further algorithm is required for inferring the optimal number of ancestry components explaining the observed data *[Rosenberg 2007]*.

There are a number of methods in order to deal with the unknown K number of ancestral populations and estimate the optimal best one from the data under analysis. In model-based methods, the algorithm is explicitly run by the user at different Ks and then the selection of the ideal K value of ancestral components is then ascertained by taking the K that optimizes the parameter of performance of the algorithm *[Wollstein and Lao 2015]*. For example:

- the one that maximizes the log-likelihood of the posterior in the case of STRUCTURE
- the one that minimizes cross-validation error is applied in ADMIXTURE
- the one that minimizes cross-entropy error is applied in SNMF

Cross Validation (ADMIXTURE) vs Cross-Entropy (SNMF)

Cross validation procedure helps identifying which value of K has the best predictive value by fitting the model on a subset of genotype data and then predicting the excluded (masked) genotypes *[Liu 2013].*

The aim of cross-validation method is to identify the best K value as judged by prediction of systematically excluded data points. In ADMIXTURE software, v-fold cross-validation procedure is performed fragmenting the non-missing genotypes into "v" more or less equally sized subsets. At each of "v" iterations, the members of one of the folds are masked (excluded temporarily marking them as missing) to build a new data matrix and then computing the log-likelihood score (the entries with missing values are ignored). Maximization of the log-likelihood readily yields new estimates for the masked data and the prediction error is estimated by averaging the squares of the deviance residuals across all masked entries over all folds. Minimizing this estimated prediction error on a grid of K values then suggests the most suitable K *[Alexander 2011].*

Cross-Entropy is a cross-validation technique also based on imputation of masked genotypes and a procedure partitioning the genotypic matrix entries into a training set and a test set. To build the test set, 5% of all genotypes are randomly selected and marked as missing values. The occurrence probabilities for the masked entries from training sets are computed according to the formula *[Frichot 2014]*:

$$p_{i\ell}^{\text{pred}}(j) = \sum_{k=1}^{K} q_{ik} g_{k\ell}(j), \quad j = 0, 1, 2$$

In statistical terms, the cross entropy method provides an estimate of the quantity:

$$H\left(p^{\text{sample}}, p^{\text{pred}}\right) = -\sum_{j=0}^{2} p^{\text{sample}}(j) \log p_{i\ell}^{\text{pred}}(j), \quad j = 0, 1, 2.$$

This quantity corresponds to the sum of the Kullback–Leiber divergence between the sampled and predicted allelic distributions. In probability theory, the Kullback–Leibler divergence is a measure of the difference between two probability distributions P and Q *[Kullback 1951].* In applications, P typically represents the "true" distribution of data or a precisely calculated distribution, while Q typically represents a theoretical model or approximation of P. Specifically, the Kullback–Leibler divergence from Q to P, is a measure of the information gained when one revises one's beliefs from the prior probability distribution Q to the posterior probability distribution P. It is the amount of information lost when Q is used to approximate P. Therefore, the number of ancestral gene clusters (K) is selected to minimize the cross-entropy criterion where smaller values of the criterion indicate better algorithm outputs and estimates.

Spatial ancestry and SPA.

As discussed previously, ancestry inference from genetic data takes populations modeled as discrete units of the input of the problem, and estimates the fractions of the genome coming from a set of source populations as the output. This inference aims to assign each allele in the genome to one of the considered ancestry populations. Alternative methodologies study population structure in a geographic continuum, exploiting the expected correlation of genetics and geography derived from isolation by distance models (again the first law of geography: *"Everything is related to everything else, but near things are more related than distant things" (Tobler, 1970))* . Such localization is called spatial ancestry assignment. This spatial approach offers some advantages *[Yang 2014]*:

i) It is reconciled with the fact that nature rarely provides precise boundaries between distinct populations which are exchanging individuals.

ii) Model-based inference take advantage of the geographic structure of allele frequencies in order to increase statistical power.

iii) More accurate allocation of ancestors for unsampled or undersampled regions.

By using this approach of ancestry inference based on geographical continuum instead of a categorical attribute, European individual's geographic coordinates of origin can be determined up to a few hundred kilometers of error using spatial ancestry inference methods. Though this level of resolution is impressive, it is natural to wonder if a model-based method for spatial assignment could perform better and whether inferences could be reliably made for admixed individuals *[Rañola and Novembre 2014].*

With the aim of validating real data from Europe, several theoretical studies using computer simulations have shown that major prehistoric demographic events can produce genetic gradients in autosomal markers similar to the observed in the real data. However, these simulations usually ignore more subtle demographic events that took place throughout history at a smaller geographical scale such as those in Europe, simplifying the demographic history due to computational constraints. For solving these obstacles, it has been suggested to pay more attention to recent demographic history in interpreting genetic clines and has been proposed that genetic population substructure is detectable on a small geographic scale despite recent demographic events *[Lao 2013]* .

In this line of analysis, SPA software is a probabilistic model for the spatial structure of genetic variation where the allele frequency of each SNP changes as a function of the location of the individual in geographic space: the allele frequency is a function of the x and y coordinates of an individual on a map. In SPA, each individual's genotypes are assumed to follow Hardy-Weinberg proportions, with allele frequencies defined by the individual's location *[Yang and Novembre 2012]*.

SPA algorithm is founded on the principle that when sampling a chromosome of an individual from a position (x,y) on the map, the probability of observing the minor allele at SNP "j" on the chromosome can be formulated assuming a spatial gradient (Figure 3.6) as follows:

$$f_j(\mathbf{x}) = \frac{1}{\exp(-\mathbf{a}_j^T \mathbf{x} - b_j) + 1}$$

where the function $f_j(X)$ is selected under the assumption to be an instance of a logistic function and being a continuous function that describes allele frequency as a function of geographic positioning. "X" is a vector of variables indicating geographic locations, "a" is a coefficient that encodes the steepness of the slope and "b" is a fixed offset parameter.

SPA model captures spatial genetic structure by the ability to jointly estimate both the allele frequency gradients and the spatial positions of individuals only from the genotype data. The software starts by placing the individuals in random positions, then iteratively uses these positions for the estimation of the slope functions, and finally using the slope functions to update the individual positions.

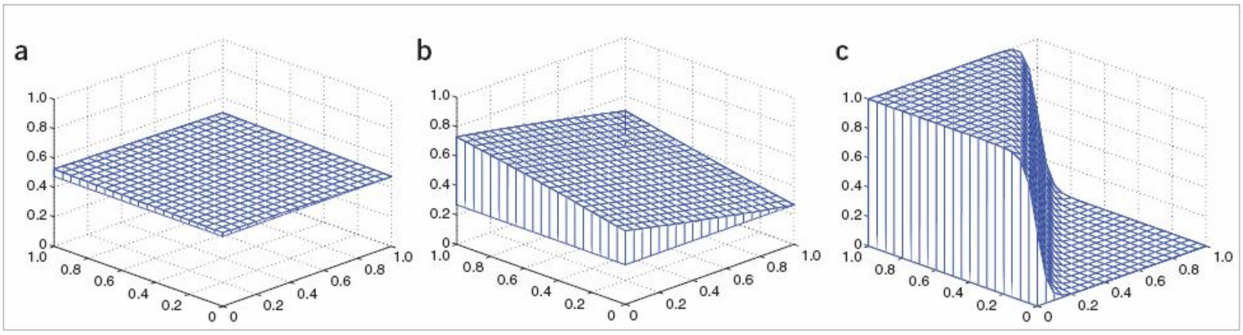


Figure 3.6: Different allele frequency slope model (a) Flat slope. A SNP with nearly constant allele frequency in all regions of the map. (b) Medium slope. A SNP with gradual allele frequency change. (c) Steep slope. A SNP with a sharp frequency change. Extracted from [A model-based approach for analysis of spatial structure in genetic data,Wen-Yun Yang, John Novembre]

4. Objectives

To what degree genetically homogeneous groups of human individuals exist is a long on-going and yet unsolved debate in the scientific community. Answering this question is important for better understanding recent human evolutionary history, for reducing the amount of false positives in gene mapping studies and other medical issues, and for inferring biogeographic origin of unknown persons in forensic investigations. Therefore, detecting the genetic fingerprint of admixture and isolation processes in human populations is of main interest for human population geneticists, and the development of new methods for detecting such events has been a constant in the literature. However, the demographic history of *Homo sapiens* has revealed to be extremely complex, comprising a large number of demographic fluctuations and migratory events that spatially and temporally overlap since the initial Out of Africa expansion of humans.

To make things more complex we need to know what is the genetic variation in *Homo Sapiens* explained by the populations. In order of importance, approximately 80% of the total genetic variability is explained by within-individual variation, a small proportion in the range 10-15% is explained by continent of origin and finally the remaining approximately 5% of the genetic variation is explained by the populations *[Wollstein and Lao 2015]*

However, as we have highlighted in the previous sections, state-of-the-art algorithms for detecting fine population substructure present several problems. First of all, even when considering the simplest demographic models, the obtained genetic admixture estimations depend on the assumptions of the algorithm, the type and number of considered DNA markers and how they were discovered initially, the underlying demographic relationship among the considered populations, and the sample size of the studied populations. When applied to real populations, most algorithms only agree at the continental level of achieved ancestry resolution, while reaching fine geographic population substructure is usually cumbersome. Furthermore, even in the simplest controlled demographic environments, the best performing algorithms show departures up to 5% in the estimated ancestry proportions of each individual compared to his true genetic ancestry *[Wollstein and Lao 2015]*. Most importantly, the behavior of these algorithms is unknown in more complex and realistic demographic models such as the ones including geography. This is particularly important in the case of humans. Humans tend to mate predominantly to individuals from the same (or close by) geographic area, which creates an effect of isolation by distance in the amount of genetic differentiation.

The aim of this project is to analyze the performance of commonly applied algorithms for detecting global ancestry in complex controlled geographic demographic scenarios in order to:

1) Establish the robustness of these algorithms
2) Identify best performing algorithms.
3) Provide guidelines for interpreting the result from these algorithms.

Demographic models will consider:

i) Two-dimensional stepping stone model for mimicking processes of isolation by distance in humans.

ii) Anisotropic models for mimicking the different continental axis of differentiation in humans.

For each demographic model, we will generate simulated full genomes by means of the demographic simulator FASTSIMCOAL2

At each simulation, we will run different commonly applied algorithms for detecting population substructure such as:

- ADMIXTURE
- SNMF
- SMARTPCA
- MDS-PLINK
- SPA

We will analyze the output performance of the different algorithms when considering non homogeneous biased geographic sampling or unequal sampling size.

5. Methods

5.1 2D stepping stone model.

We have depicted in the next figure a two dimensional stepping stone model of 225 demes on a grid of 15x15. Each population can exchange migrants with a rate “m” per generation with the nearby neighbor surrounding populations. The figure is a simple graph illustrating the considered 2D stepping stone in a 15 x 15 grid where each population is depicted as a circle. Allowed migrations are depicted as edges. Each number represents one population. This is one of the two demographic models that are the object of this study.

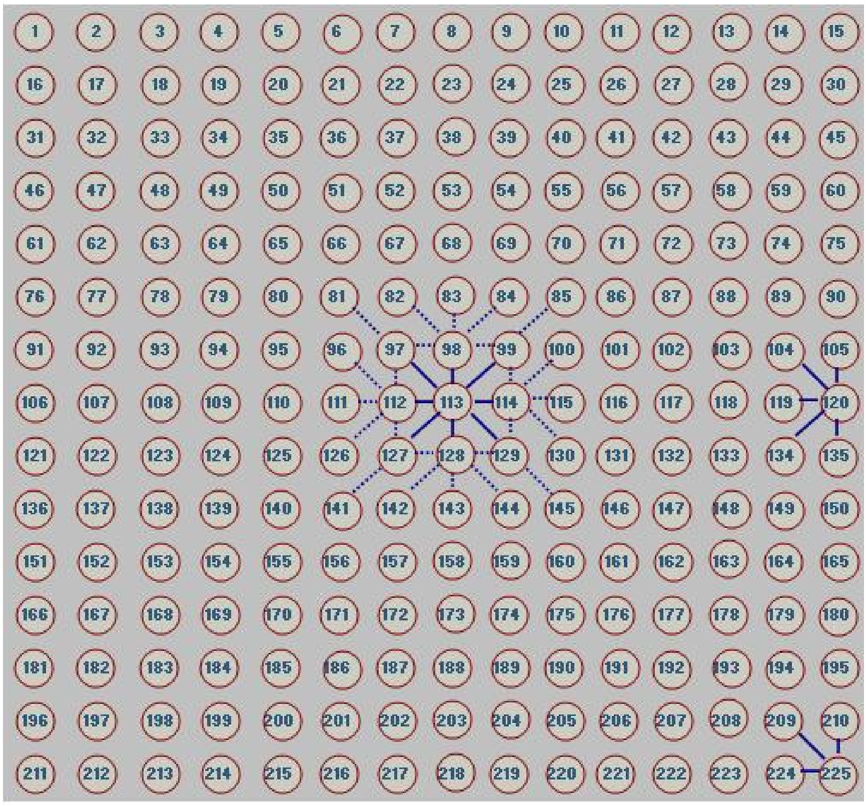


Figure 5.1: Two Stepping Stone Model, 225 demes, 15x15. Demes 113, 120 and 225 illustrate migration patterns between neighbouring demes, patterns which in fact are applied in all demes in the same manner but not painted in here.

5.2 Anisotropic model.

We have depicted in the next figure an anisotropic model of 125 demes on a cross-shaped grid with five blocks of 5x5. Cross-shaped grid emulates anisotropic behavior. Each population can exchange migrants with a rate "m" per generation with the nearby neighbor surrounding populations. Again, each population is depicted as a circle, allowed migrations are depicted as edges and each number represents one population.

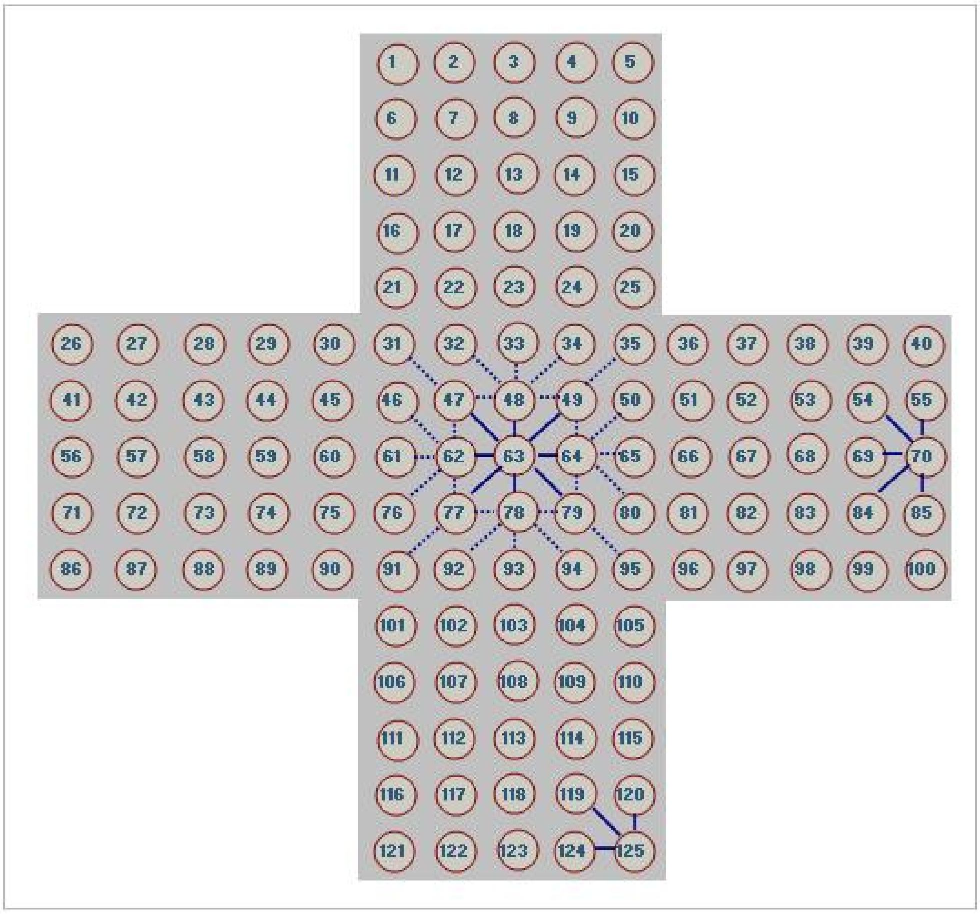


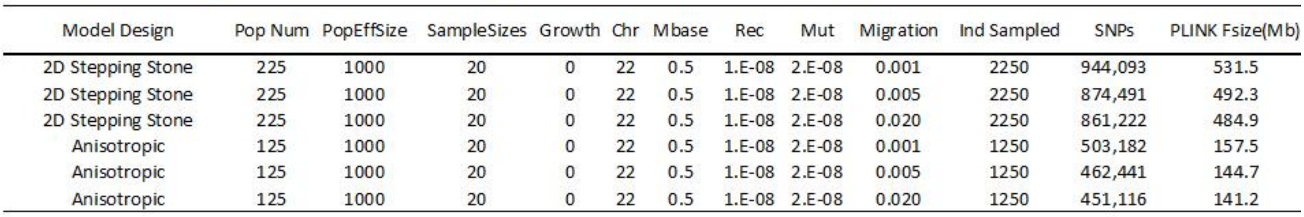
Figure 5.2: Anisotropic Model, 125 demes, cross-shaped grid with 5 blocks of 5x5 emulating directionally dependent behavior. Demes 63, 70 and 125 illustrate migration patterns between neighbouring demes, patterns which in fact are applied in all demes in the same manner but not painted in here.

5.3 Demographic simulations with FastSIMCoal2.

We have run Fastsimcoal2 six times to simulate 2D Stepping Stone and Anisotropic models. ,Each scenario has been run with three different migration rates: m=0.001, m=0.005 and m=0.02. Taking as example m=0.001 and the population numbered as 113 in Figure 5.1 for the 2D Stepping Stone model, for each generation backward in time, any gene from population 113 has probability 0.001 to be sent to populations {97,98,99,112,114,127,128,129} and that a gene from populations {97,98,99,112,114,127,128,129} has a probability 0.001 to move to population 113. This migration process is applied to all demes in the model following a migration matrix defined in the fastsimcoal2 input file “PAR”: the migration matrix included in the PAR input file is a double entry matrix of dimension 225x225 for the 2D stepping stone model and 125x125 for the anisotropic model, and the cell values of this matrix are zero for those pairs of demes without migration or 0.001/0.005/0.02 for those pairs of nearby neighbouring demes as defined in Figure 5.1 and Figure 5.2.

The fixed parameters common for the six models are: Population Effective Size=1000 haploid individuals (or 500 diploid individuals) for each of the 225 populations in the 2D stepping stone design (and also for each of the 125 populations in the anisotropic design), Sample Size=20 haploid individuals, Growth=0 that means demes have a stationary population size with no expansion events, and we want to generate diversity along 500 Kb DNA sequence on 22 Chromosomes with fixed mutation rate $\mu = 2 \times 10^{-8}$ /bp/gen and fixed recombination rate $\mu = 1 \times 10^{-8}$ /bp/gen. These demographic parameters have been ascertained either due to computational constraints (i.e. number of demes, effective population size, DNA fragment size) or because they represent real case scenarios in human populations (i.e. sample size, number of chromosomes, recombination rate and mutation rate). Summary of the parameters established in the six PAR input files used to run fastsimcoal2 are shown in Table 5.1.

Table 5.1: fastsimcoal2 parameters for the two demographic models and the three levels of migration rate.



Model Design	Pop Num	PopEffSize	SampleSizes	Growth	Chr	Mbase	Rec	Mut	Migration	Ind Sampled	SNPs	PLINK Fsize(Mb)
2D Stepping Stone	225	1000	20	0	22	0.5	1.E-08	2.E-08	0.001	2250	944,093	531.5
2D Stepping Stone	225	1000	20	0	22	0.5	1.E-08	2.E-08	0.005	2250	874,491	492.3
2D Stepping Stone	225	1000	20	0	22	0.5	1.E-08	2.E-08	0.020	2250	861,222	484.9
Anisotropic	125	1000	20	0	22	0.5	1.E-08	2.E-08	0.001	1250	503,182	157.5
Anisotropic	125	1000	20	0	22	0.5	1.E-08	2.E-08	0.005	1250	462,441	144.7
Anisotropic	125	1000	20	0	22	0.5	1.E-08	2.E-08	0.020	1250	451,116	141.2

<u>OUTPUT FILES from fastsimcoal2</u>

For each simulated dataset we obtained an output text file in the Arlequin format (ARP). The content of each ARP file is formed by four blocks of information:

i) A header listing the run parameters used.

ii) A block listing the numerical positions of the generated SNPs on each of the 22 chromosomes. One row per chromosome with the exact locations separated by commas. For instance, for a given simulation, the block starts with chromosome-1:

```
# 20675 polymorphic positions on chromosome 1
#10, 37, 51, 63, 152, 171, 185, 214, 232, 241, 248, 311, 328, 387, 433, 443, 488, 490,
494, 535, 539, 574, 598, 638, 653, 668, 737, 741, 754, 798, 807, 870, 893, 915, 930, 1004,
……………………………………, 499672, 499690, 499785, 499795, 499931, 499943, 499945, 499950
```

and ends with the list of SNPs generated on chromosome 22:

```
# 20625 polymorphic positions on chromosome 22
#2, 13, 28, 56, 100, 111, 147, 153, 167, 184, 208, 232, 238, 391, 420, 426, 435, 437, 471,
488, 536, 543, 576, 942,......................, 499699, 499756, 499817, 499878, 499892,
499897, 499912, 499919, 499921, 499937, 499961
```

iii) A block listing the numerical positions of the recombinations events generated on each of the 22 chromosomes. One row per chromosome similar to prior SNPs lists.

iv) A block with the DNA sequence simulated for each haploid individual: for each haploid individual and each of the defined positions (those positions provided above in block [ii]), it is provided the allele. Fastsimcoal2 only outputs polymorphic sites for DNA sequences, unless the user request to output all sites by using a command line option.

For example, for the first haploid individual (individual coded 1_1) the allele corresponding to the first SNP (in position 10) is "A":

```
1_1   1
AGAAATCCCTTAGCTATCAGGATCATGGGGAGTCCGCGAAGGTGGATTGTATCCCAGATAAGTGTGCGCACCCGATTGATAACGGTTAG
GGCCGCGATGGCTCGGAGGGAAGGTGCACATCAACACCTATCTCTCGGCCGCCGGTCAATGGATCGATTCACGATCCAAGATAATAGGG
GGTCCCTCACGGCGCCAATAGAT…………………...
```

The six ARP Arlequin files showed in Table 5.1 have been converted to PLINK BED format files by using a Java application programmed for that specific purpose (ConvertArlequinToPlink.jar).

5.4 Sampling methods.

Spatial Sampling.

We have considered various sampling strategies for the spatial distribution of the selected demes and for the sampling size of individuals by population.

For the spatial sampling we have considered three scenarios:

- Full sampling (Homogeneous): taking the six basic designs (Table1) we consider all the simulated populations (225 and 125 for the 2D stepping stone and anisotropic models respectively) with all of original individuals per population (2250 and 1250 diploid individuals for 2D stepping stone and anisotropic models respectively). Just as they are conceived after running fastsimcoal2. We consider these six scenarios as main points of reference, being null models for testing each method.
- Random spatial sampling: from the full sampling we have selected "k" populations at random: k=75 for 2D stepping stone and k=45 for anisotropic model. This represents a similar percentage of sampling 33% and 36% respectively. Figure 5.3 shows the random selection on the 2D stepping stone and anisotropic designs.

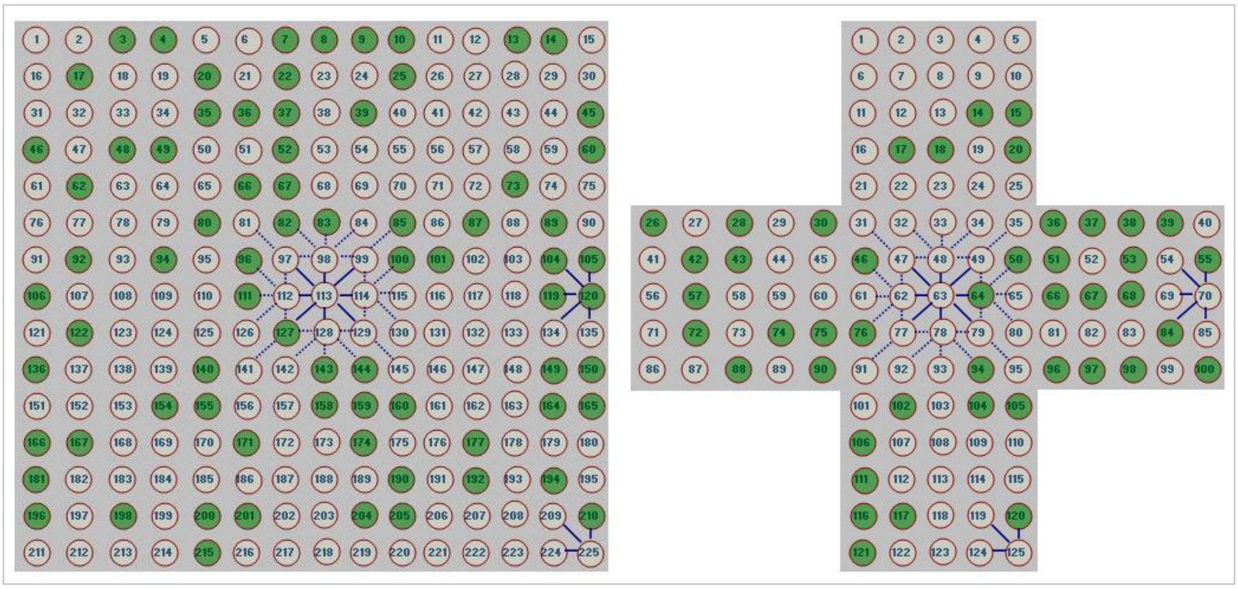


Figure 5.3: The k=75 and k=45 random populations sampled from the total 225 populations of the 2D stepping stone model full model and from the total 125 of anisotropic model.

- Contagious sampling: The term contagious distribution was apparently first used by Neyman (1939) for a discrete distribution that exhibits clustering or contagious effect. In plant ecology, contagious distribution appears when the pattern formed by the distribution of individuals of a given plant species within a community is not random but shows clumping. In our case, from the full sampling, we have declared "n" regions and in each region we have selected "p" populations. Both values, "n" and "p" have been chosen in a manner which ensures $n \times p \approx k$ for comparison purposes. For 2D stepping stone models we have declared n=25 squared regions formed by 9 populations each one. For every of these 25 regions we have selected 3 populations in average per region (total 74 populations similar to the 75 random selection). For anisotropic models we have declared n=5 squared regions formed by 25 populations each one. For every of these 5 regions we have selected 9 populations per region (total 45 populations equal to the random selection). Figure 5.4 shows the contagious sampling for the two demographic strategies.

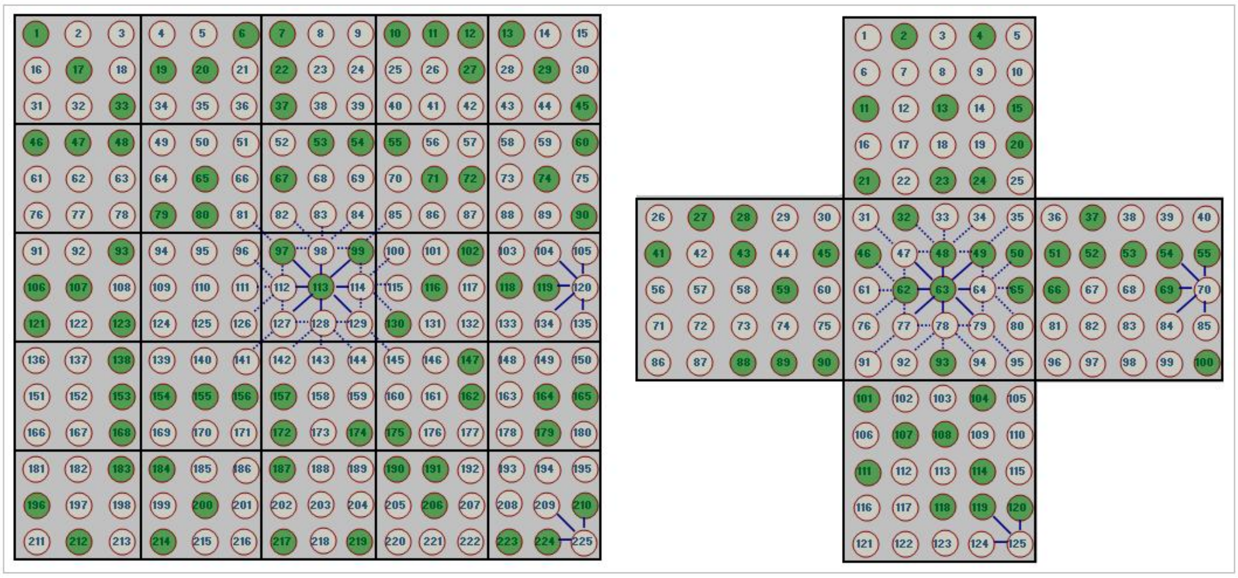


Figure 5.4: Contagious sampling showing the 25 regions declared on 2D stepping stone (left) and the 5 regions on Anisotropic model(right). 3 populations per region in average have been selected on 2D stepping stone model, total 74 (left) and 9 populations per region on Anisotropic (right).

<u>Sampling size by population.</u>

Regarding the sampling size by population we have considered two options: the same number of individuals per population (10 diploid) and unequal number of individuals per population defined by a random sampling within populations in the range 1 to 5 diploid individuals.

<u>Minimum allele Frequency (MAF) and Linkage Disequilibrium (LD) filtering.</u>

In addition, each of the subsets generated by prior sampling strategies (spatial and within populations) have been subject to Minimum allele Frequency (MAF) and Linkage Disequilibrium (LD) filtering. The MAF filtering has been performed by including only those SNPs that are above an specific MAF value (0.05), which is typical in GWAS and SNP microarray platforms. The LD filtering has been implemented based on the variance inflation factor (VIF), which recursively removes SNPs within a sliding window (window size used in SNPs = 50), the number of SNPs to shift the window at each step (value used = 5) and the VIF threshold (value used VIF=2). The VIF is equal to 1/(1-R^2) where R^2 is the multiple correlation coefficient for a SNP being regressed on all other SNPs simultaneously. That is, this considers the correlations between SNPs but also between linear combinations of SNPs. A VIF=2 implies R^2=0.5. These are the default LD pruning parameters in PLINK.

Both pruning actions, MAF and LD, have been implemented via PLINK software.

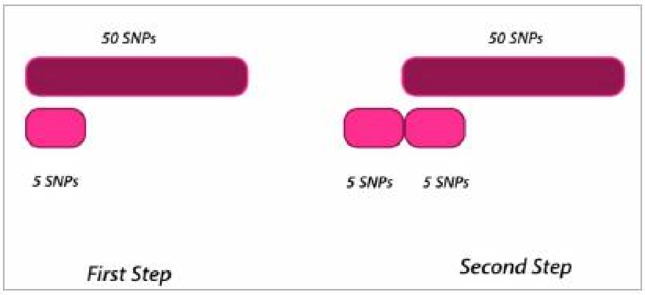


Figure 5.5: LD-based SNP pruning: generates a subset of SNPs that are in approximate LD. Sliding window 50 SNPs and calculate LD. Select representative SNPs which have low LD (R^2< 0.5).

<u>Sampling Summary.</u>

Recapitulating and bearing in mind the six starting full models from the two demographic designs with three different migration rates (Table 5.1), each one of them have generated a total of 12 additional subsets (2x2x3): 2 variants due to random or contagious spatial population sampling, 2 variants due to within equal or unequal population sampling, 3 variants due to non filtering or MAF or LD filtering.

In total we have built 78 experimental scenarios and each one is represented by its PLINK-BED file, as shown in Table 5.2A and Table 5.2B.

Table 5.2A: Experimental Dataset Generation for 2D Stepping Stone models

	Demographic	Migration	Population	Num	Individual	Num	Num	plink-bed	MAF	LD
#	Model	Rate (m)	Sampling	Pops	Sampling	Inds	SNPs	Fsize(Mb)	Filtering	Filtering
1	2D S.Stone	0.001	Full model	225	Equal	2250	944,093	531.5	No	No
2	2D S.Stone	0.005	Full model	225	Equal	2250	874,491	492.3	No	No
3	2D S.Stone	0.020	Full model	225	Equal	2250	861,222	484.9	No	No
4	2D S.Stone	0.001	Random	75	Equal	750	944,668	177.6	No	No
5	2D S.Stone	0.001	Random	75	Equal	750	328,919	61.8	0.05	No
6	2D S.Stone	0.001	Random	75	Equal	750	440,162	82.8	No	R2=0.5
7	2D S.Stone	0.001	Random	75	Unequal	225	944,668	53.8	No	No
8	2D S.Stone	0.001	Random	75	Unequal	225	327,575	18.7	0.05	No
9	2D S.Stone	0.001	Random	75	Unequal	225	337,558	19.2	No	R2=0.5
10	2D S.Stone	0.001	Contagious	74	Equal	740	944,668	174.8	No	No
11	2D S.Stone	0.001	Contagious	74	Equal	740	328,454	60.8	0.05	No
12	2D S.Stone	0.001	Contagious	74	Equal	740	442,412	81.8	No	R2=0.5
13	2D S.Stone	0.001	Contagious	74	Unequal	211	944,668	50.1	No	No
14	2D S.Stone	0.001	Contagious	74	Unequal	211	325,174	17.2	0.05	No
15	2D S.Stone	0.001	Contagious	74	Unequal	211	334,111	17.7	No	R2=0.5
16	2D S.Stone	0.005	Random	75	Equal	750	874,975	164.5	No	No
17	2D S.Stone	0.005	Random	75	Equal	750	296,178	55.7	0.05	No
18	2D S.Stone	0.005	Random	75	Equal	750	422,053	79.3	No	R2=0.5
19	2D S.Stone	0.005	Random	75	Unequal	225	874,975	49.9	No	No
20	2D S.Stone	0.005	Random	75	Unequal	225	295,573	16.8	0.05	No
21	2D S.Stone	0.005	Random	75	Unequal	225	314,910	17.9	No	R2=0.5
22	2D S.Stone	0.005	Contagious	74	Equal	740	874,975	161.9	No	No
23	2D S.Stone	0.005	Contagious	74	Equal	740	296,679	54.9	0.05	No
24	2D S.Stone	0.005	Contagious	74	Equal	740	421,628	78.0	No	R2=0.5
25	2D S.Stone	0.005	Contagious	74	Unequal	211	874,975	46.4	No	No
26	2D S.Stone	0.005	Contagious	74	Unequal	211	294,351	15.6	0.05	No
27	2D S.Stone	0.005	Contagious	74	Unequal	211	309,956	16.4	No	R2=0.5
28	2D S.Stone	0.020	Random	75	Equal	750	861,222	161.9	No	No
29	2D S.Stone	0.020	Random	75	Equal	750	290,521	54.6	0.05	No
30	2D S.Stone	0.020	Random	75	Equal	750	424,465	79.8	No	R2=0.5
31	2D S.Stone	0.020	Random	75	Unequal	225	861,222	49.1	No	No
32	2D S.Stone	0.020	Random	75	Unequal	225	290,163	16.5	0.05	No
33	2D S.Stone	0.020	Random	75	Unequal	225	314,317	17.9	No	R2=0.5
34	2D S.Stone	0.020	Contagious	74	Equal	740	861,222	159.3	No	No
35	2D S.Stone	0.020	Contagious	74	Equal	740	290,841	53.8	0.05	No
36	2D S.Stone	0.020	Contagious	74	Equal	740	421,952	78.1	No	R2=0.5
37	2D S.Stone	0.020	Contagious	74	Unequal	211	861,222	45.6	No	No
38	2D S.Stone	0.020	Contagious	74	Unequal	211	288,025	15.3	0.05	No
39	2D S.Stone	0.020	Contagious	74	Unequal	211	309,480	16.4	No	R2=0.5

Table 5.2B: Experimental Dataset Generation for Anisotropic models

Demographic		Migration	Population	Num	Individual	Num	Num	plink-bed	MAF	LD
#	Model	Rate (m)	Sampling	Pops	Sampling	Inds	SNPs	Fsize(Mb)	Filtering	Filtering
40	Anisotropic	0.001	Full model	125	Equal	1250	503,182	157.5	No	No
41	Anisotropic	0.005	Full model	125	Equal	1250	462,441	144.7	No	No
42	Anisotropic	0.020	Full model	125	Equal	1250	451,116	141.2	No	No
43	Anisotropic	0.001	Random	45	Equal	450	503,390	56.9	No	No
44	Anisotropic	0.001	Random	45	Equal	450	186,436	21.1	0.05	No
45	Anisotropic	0.001	Random	45	Equal	450	213,424	24.1	No	R2=0.5
46	Anisotropic	0.001	Random	45	Unequal	143	503,390	18.1	No	No
47	Anisotropic	0.001	Random	45	Unequal	143	185,538	6.7	0.05	No
48	Anisotropic	0.001	Random	45	Unequal	143	159,830	5.8	No	R2=0.5
49	Anisotropic	0.001	Contagious	45	Equal	450	428,046	48.4	No	No
50	Anisotropic	0.001	Contagious	45	Equal	450	186,857	21.1	0.05	No
51	Anisotropic	0.001	Contagious	45	Equal	450	216,714	24.5	No	R2=0.5
52	Anisotropic	0.001	Contagious	45	Unequal	137	503,390	17.6	No	No
53	Anisotropic	0.001	Contagious	45	Unequal	137	186,885	6.5	0.05	No
54	Anisotropic	0.001	Contagious	45	Unequal	137	158,802	5.6	No	R2=0.5
55	Anisotropic	0.005	Random	45	Equal	450	462,587	52.3	No	No
56	Anisotropic	0.005	Random	45	Equal	450	164,598	18.6	0.05	No
57	Anisotropic	0.005	Random	45	Equal	450	207,715	23.5	No	R2=0.5
58	Anisotropic	0.005	Random	45	Unequal	143	462,587	16.7	No	No
59	Anisotropic	0.005	Random	45	Unequal	143	163,384	5.9	0.05	No
60	Anisotropic	0.005	Random	45	Unequal	143	151,457	5.5	No	R2=0.5
61	Anisotropic	0.005	Contagious	45	Equal	450	462,587	52.3	No	No
62	Anisotropic	0.005	Contagious	45	Equal	450	164,530	18.6	0.05	No
63	Anisotropic	0.005	Contagious	45	Equal	450	209,720	23.7	No	R2=0.5
64	Anisotropic	0.005	Contagious	45	Unequal	137	462,587	16.2	No	No
65	Anisotropic	0.005	Contagious	45	Unequal	137	164,583	5.8	0.05	No
66	Anisotropic	0.005	Contagious	45	Unequal	137	149,677	5.2	No	R2=0.5
67	Anisotropic	0.020	Random	45	Equal	450	451,116	51.0	No	No
68	Anisotropic	0.020	Random	45	Equal	450	160,374	18.1	0.05	No
69	Anisotropic	0.020	Random	45	Equal	450	207,884	23.5	No	R2=0.5
70	Anisotropic	0.020	Random	45	Unequal	143	451,116	16.2	No	No
71	Anisotropic	0.020	Random	45	Unequal	143	158,862	5.7	0.05	No
72	Anisotropic	0.020	Random	45	Unequal	143	148,636	5.4	No	R2=0.5
73	Anisotropic	0.020	Contagious	45	Equal	450	451,116	51.0	No	No
74	Anisotropic	0.020	Contagious	45	Equal	450	160,560	18.1	0.05	No
75	Anisotropic	0.020	Contagious	45	Equal	450	208,853	23.6	No	R2=0.5
76	Anisotropic	0.020	Contagious	45	Unequal	137	451,116	15.8	No	No
77	Anisotropic	0.020	Contagious	45	Unequal	137	160,477	5.6	0.05	No
78	Anisotropic	0.020	Contagious	45	Unequal	137	147,651	5.2	No	R2=0.5

5.5 Statistics for comparing inferred ancestry with coordinates and ancestry proportions.

Procrustes Test

In statistic terms, Procrustes analysis determines a linear transformation of the points in matrix Y to best conform them to the points in matrix X. The transformation includes translation, reflection, orthogonal rotation, and scaling. The goodness-of-fit criterion is the sum of squared errors. Procrustes algorithm returns the minimized value of this dissimilarity measure.

In general terms, Procrustes transformation compare the shapes of two or more objects that must be first optimally overlapped or superimposed. Procrustes superimposition is performed by optimally translating, rotating and uniformly scaling the objects. Both the placement in space and the size of the objects are freely adjusted. The aim is to obtain a similar placement and size, by minimizing a measure of shape difference called the Procrustes distance between the objects.

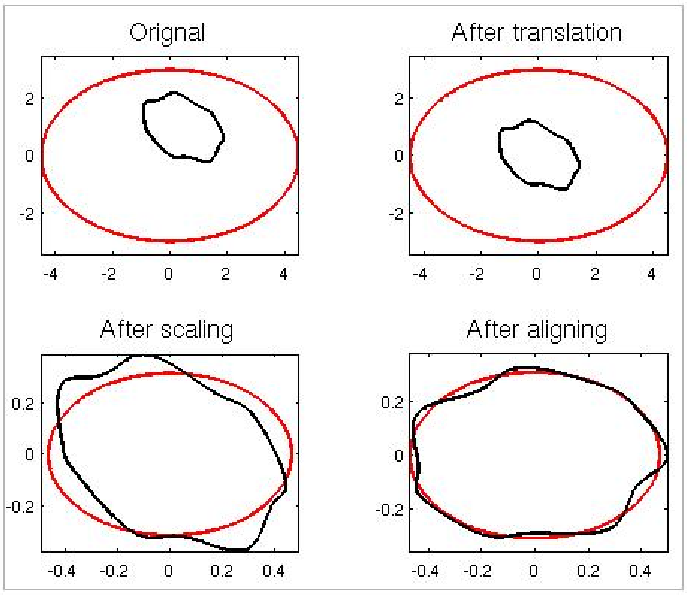


Figure 5.6: Procrustes transformation steps applied to a simple visual pair of objects.

For our study, Procrustes transformation is useful for comparisons between two or more maps that involve population-genetic data. This kind of analysis has generally been assessed in a qualitative manner, by visual evaluation. Procrustes method provides a sensible quantitative approach for map comparison: each of two maps is transformed, preserving relative distances among pairs of points within each map *[Wang 2010]*. The objective is to identify the transformations that maximize the similarity of the transformed maps and obtain the similarity score between the two optimally transformed maps. A permutation test can then evaluate the probability that a randomly chosen permutation of the points in one of the maps leads to a greater similarity score than that observed for the actual data points.

More formally, Procrustes method aims to find the transformations, f* and g*, that minimize a function $d(f(X), g(Y))$ over all choices f and g that preserve relative pairwise distances between points in X and Y. Being both X and Y a couple of $n \times k$ matrices, only X is transformed so g*(Y) = Y can be assumed. The transformation f can be written as $f(x_r) = \rho A^T x_r + b$, where ρ is a scalar to produce matrix dilation, A is a k × k orthogonal matrix representing a rotation and possibly a reflection, and b is a k × 1 translation vector.

The objective function “d” to be minimized is as follows:

$$d(f(\mathrm{X}), \mathrm{Y}) = \sum_{r=1}^{n} (\mathrm{y}_r - f(\mathrm{x}_r))^T (\mathrm{y}_r - f(\mathrm{x}_r))$$

For the 78 experimental scenarios defined above (Tables 2A-2B), we have sequentially performed the algorithm free methods PCA and MDS as well as the model-based algorithm SPA. The three methods return a two columns matrix with the estimated coordinates of the N individuals (rows) present in the corresponding 78 PLINK-BED files described before.

We have applied Procrustes method using function "protest" from R package "vegan" which rotates a matrix to maximum similarity with a target matrix minimizing sum of squared differences. This function has been recursively executed to obtain the Procrustes correlation coefficient returned when comparing each of the 78x3 (PCA,MDS,SPA) coordinates matrices with the real geographical positions of the individuals: we have used the algorithms PCA,MDS and SPA to solve a problem for which we know the answer in advance. The three programs have estimated the geographical coordinates of the individuals (78 times) and we can compare their results with the real coordinates of the demographical models as described in Figure 5.1 and Figure 5.2. For instance, for the 2D stepping stone model, individuals belonging to population number 50 have real geographical coordinates (5,4), individuals belonging to population 100 have coordinates (10,7), and so on. Same simplicity for anisotropic model.

Mantel Test

The Mantel test is a non-parametric statistical method that computes the correlation between two distance matrices. The Mantel test was proposed in 1967 to test the association between two matrices and was first applied in population genetics by Sokal in 1979. It computes the significance of the correlation through permutations of the rows and columns of one of the input distance matrices. The test statistic is the Pearson product-moment correlation coefficient "r" which falls in the range of -1 to +1, where being close to -1 indicates strong negative correlation and +1 indicates strong positive correlation. An r value of 0 indicates no correlation.

Formally the Mantel test is given by:

$$Z_m = \sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij} \times d_{ij}$$

where g_{ij} and d_{ij} are the two distance matrices to be evaluated *[Diniz-Filho 2013]*. Because Zm is given by the sum of products of distances its value depends on how many populations are studied, as well as the magnitude of their distances. The Zm-value can be compared with a null distribution, and Mantel originally proposed to test it by the standard normal deviate (SND), given by

$$\mathrm{SND} = Z_m/\mathrm{var}(Z_m)^{1/2}$$

The rationale behind Mantel Test is that if there is a relationship between matrices G and D, the sum of products Zm will be relatively high, and randomizing rows and columns will destroy this relationship so that Zm values, after permutations, will tend to be lower than the observed *[Diniz-Filho 2013]*. Therefore if the null hypothesis of there being no relation between the two matrices is true, then permuting the rows and columns of the matrix should be equally likely to produce a larger or a smaller coefficient. In contrast to the ordinary use of the correlation coefficient, in Mantel test correlation is recalculated after each permutation. The p-value of the observed correlation is the proportion of such permutations that lead to a higher correlation coefficient.

We have converted the 78x3 (PCA,MDS,SPA) coordinate matrices and the real geographical positions of the individuals into Euclidean distance matrices. Then, function "mantel" from R package "ade4" has been recursively executed to obtain the correlation coefficient returned when comparing each of the 78x3 (PCA,MDS,SPA) distance matrices with the distance matrix from real geographical positions. Similar procedural approach as before with Procrustes test: we are assessing which of the experimental designs reconciles better with the real geographical coordinates.

CLUMPP

As discussed above, several algorithms for inferring ancestry proportions such as ADMIXTURE or SNMF result on a matrix where for each individual (rows) is given a membership coefficient for each cluster or ancestry populations (columns), being these coefficients the ancestry fractions (probabilities) assigned to every individual and summing to 1 across the K columns. The random initial conditions of these clustering algorithms introduce a degree of randomness in the output results, and independent analyses of the same input data may result in several distinct outputs. According to *[Rosenberg 2007]* the main differences across replicates are of two types: “label switching” and “genuine multimodality”. “Label switching” refers to the case in which different replicates obtain the same numerical ancestry fractions but placed in different columns (clusters are permuted). Due to the meaning of each cluster label is not known in advance, a clustering algorithm may be equally likely to reach any of K! permutations of the same collection of estimated membership coefficients. In contrast to label switching, “genuine multimodality” appears when different admixture scenarios can similarly explain the observed genetic diversity in the data.

These matrices with ancestry fractions from multiple runs of a clustering program are the input for CLUMPP program, which outputs these same matrices, permuted so that all replicates have as close as possible a match. CLUMPP resolves the “label switching” heterogeneity so that the “genuine multi-modality’ can be detected and quantified.

Having a C×K matrix of membership coefficients for a single cluster analysis where “C” are the individuals or populations and the “K” columns correspond to clusters, and R replicates of the admixture analysis, CLUMPP attempts to maximize a measure of similarity of the replicates of the original CxK matrix over all $(K!)^{R-1}$ possible alignments of the replicates. The coefficient G’ used by CLUMPP to measure the similarity is defined as follows:

$$\boxed{G'(Q_i, Q_j) = 1 - \frac{\|Q_i - Q_j\|_F}{\sqrt{2C}}}$$

Where Qi and Qj are a couple of input matrices coming from runs “i” and “j” and we calculate Frobenius norm on their difference matrix Qi-Qj:

$$\boxed{\|A\|_F = \sqrt{\sum_{c=1}^{C}\sum_{k=1}^{K} a_{ck}^2}}$$

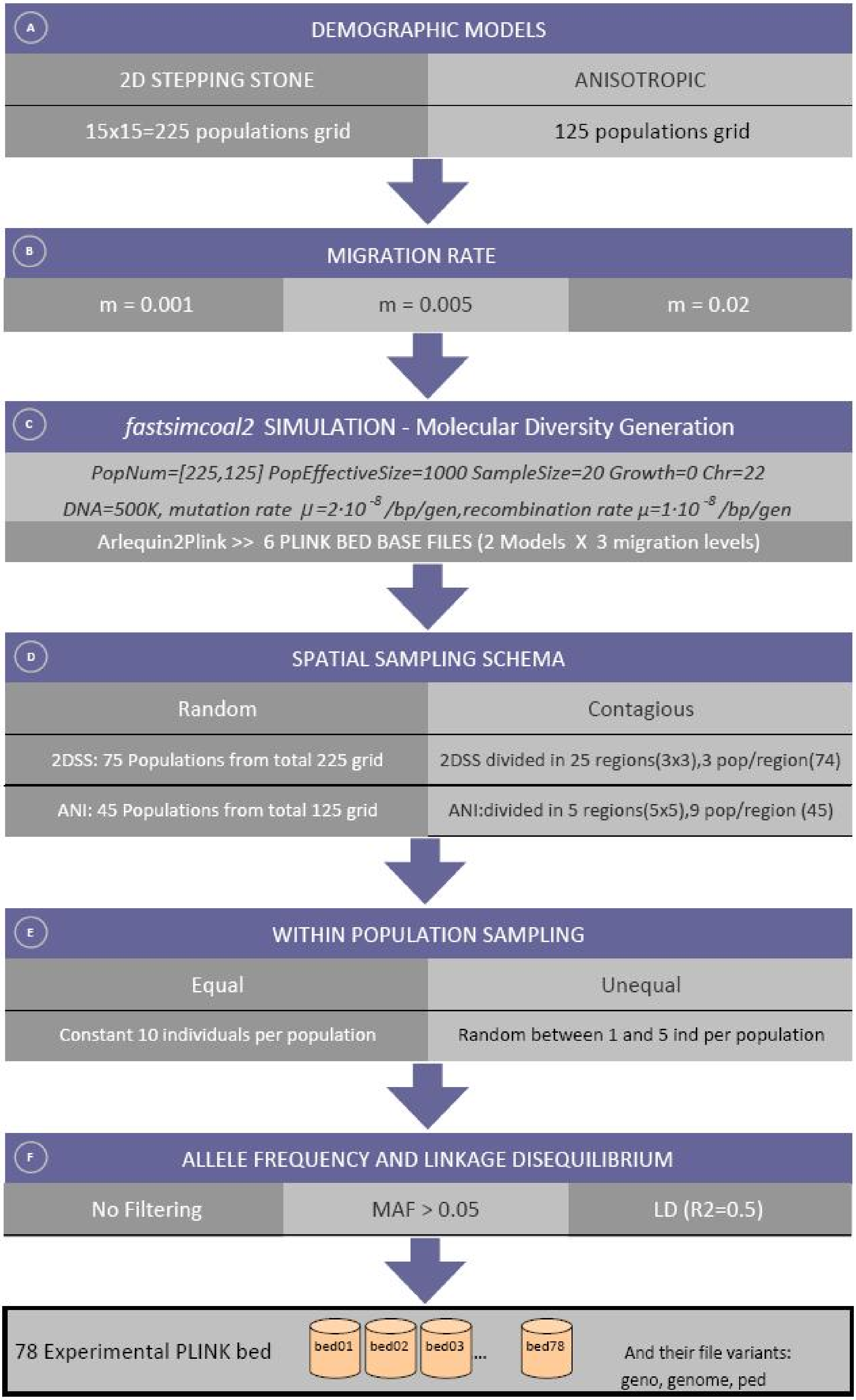
The output coefficient G’ is a value in the range [0,1] and the maximum G’=1 corresponds to an identical pair of input matrices, decreasing G’ as the similarity of the input matrices decreases.

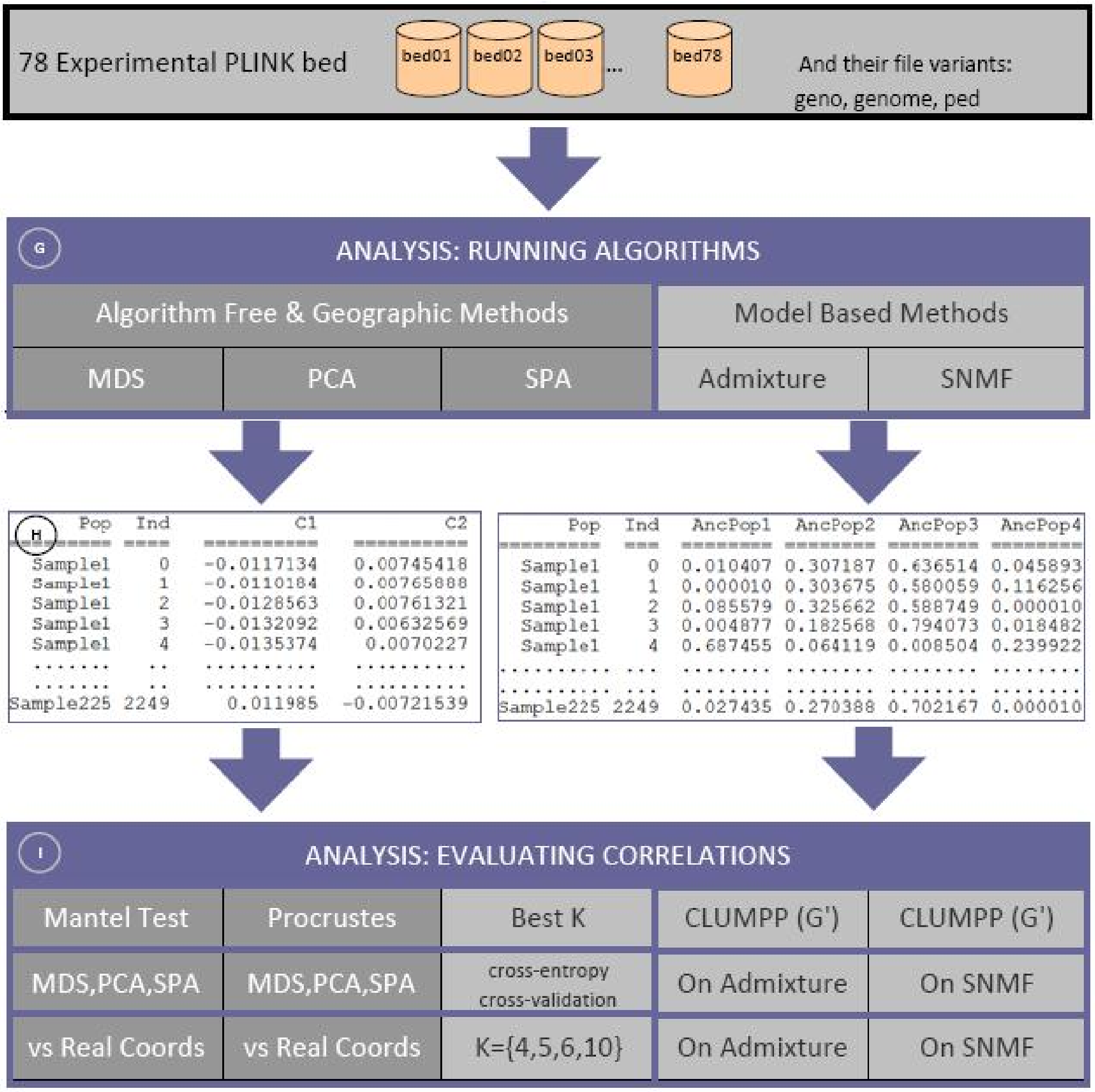
CLUMPP program has been used to evaluate the degree of similarity between the matrices obtained from the 78 runs of ADMIXTURE and SNMF algorithms. In particular, we have compared by blocks the resulting matrices from sampling runs against their corresponding full model matrices: the resulting matrices from ADMIXTURE and SNMF on 2D stepping stone full model basic case (migration rate m=0.001) with all the sampling cases generated on this basis (12 cases). The same for the other five blocks: 2D stepping stone/migration rate 0.005 and 0.02 and anisotropic models with the three migration rates variants.

6. Results

6.1 Experimental Workflow.

We have constructed an experimental model for testing the performance of currently algorithms applied for estimating population substructure which starts by designing two ideal prototypes of spatially structured populations (2D stepping stone and anisotropic). From each model we have generated a pool of 78 experimental datasets, simulating the genomic molecular diversity with Fastsimcoal2, performing the sampling of individuals and populations and selecting different filtering strategies (MAF, LD). Those 78 datasets (plink bed files) have been processed to evaluate the response of commonly applied algorithms to SNP data for quantifying individual population substructure: PCA, MDS, SPA, ADMIXTURE and SNMF. For those algorithms in which the output is a coordinate (PCA, MDS and SPA), we have evaluated the correlation (via Mantel and Procrustes tests) of these estimated coordinates with the geographic sampling coordinates of individuals in our original ideal artifacts. For ADMIXTURE and SNMF we have applied different algorithms for assessing the best K number of ancestries and we have applied CLUMPP software to compare their output matrices. Figure JJ describes the work-pipeline that we have applied.


A
DEMOGRAPHIC MODELS
2D STEPPING STONE
ANISOTROPIC
15x15=225 populations grid
125 populations grid
B
MIGRATION RATE
m = 0.001
m = 0.005
m = 0.02
C
fastsimcoal2 SIMULATION - Molecular Diversity Generation
PopNum=[225,125] PopEffectiveSize=1000 SampleSize=20 Growth=0 Chr=22
DNA=500K, mutation rate $\mu=2\cdot10^{-8}$/bp/gen,recombination rate $\mu=1\cdot10^{-8}$/bp/gen
Arlequin2Plink >> 6 PLINK BED BASE FILES (2 Models X 3 migration levels)
D
SPATIAL SAMPLING SCHEMA
Random
Contagious
2DSS: 75 Populations from total 225 grid
2DSS divided in 25 regions(3x3),3 pop/region(74)
ANI: 45 Populations from total 125 grid
ANI:divided in 5 regions(5x5),9 pop/region (45)
E
WITHIN POPULATION SAMPLING
Equal
Unequal
Constant 10 individuals per population
Random between 1 and 5 ind per population
F
ALLELE FREQUENCY AND LINKAGE DISEQUILIBRIUM
No Filtering
MAF > 0.05
LD (R2=0.5)
78 Experimental PLINK bed
bed01
bed02
bed03
...
bed78
And their file variants:
geno, genome, ped



```
       Pop   Ind           C1           C2
==========  ====  ===========  ===========
   Sample1     0   -0.0117134   0.00745418
   Sample1     1   -0.0110184   0.00765888
   Sample1     2   -0.0128563   0.00761321
   Sample1     3   -0.0132092   0.00632569
   Sample1     4   -0.0135374    0.0070227
   .......    ..  ...........  ...........
   .......    ..  ...........  ...........
Sample225  2249     0.011985  -0.00721539
```

```
       Pop  Ind  AncPop1  AncPop2  AncPop3  AncPop4
==========  ===  ======== ======== ======== ========
   Sample1    0  0.010407 0.307187 0.636514 0.045893
   Sample1    1  0.000010 0.303675 0.580059 0.116256
   Sample1    2  0.085579 0.325662 0.588749 0.000010
   Sample1    3  0.004877 0.182568 0.794073 0.018482
   Sample1    4  0.687455 0.064119 0.008504 0.239922
..........  ...  ........ ........ ........ ........
..........  ...  ........ ........ ........ ........
Sample225 2249  0.027435 0.270388 0.702167 0.000010
```

Figure 6.1: Workflow for the full experimental procedure detailing the two main stages of the pipeline: First, dataset pool generation by simulation and sampling from step [A] to [F] and second, the analysis and obtention of results from step [G] to [I].

[A] Generation of the two demographic models which are the basis of the analysis as described in Figure 5.1 (2D stepping stone) and Figure 5.2 (anisotropic), a squared grid of 15x15(225) populations and a cross-shaped grid with 125 populations respectively.

[B] Our design introduces three levels of migration rate applied to each of demographic models to evaluate the impact of migration degree on final results.

[C] Prior steps [A] and [B] are just conceptual and graphical: the models do not become computational artifacts until we build the parameter file for fastsimcoal2 . By completing the input file (PAR file) we are defining how we want to simulate the molecular diversity through fastsimcoal2 (populations, mutation rate, recombination , etc).

[D] After 6 runs of fastsimcoal2 and once the six Arlequin result files have been converted to plink-BED format, spatial sampling schema is applied by generating subsets based on random and contagious distributions as shown in Figure DD and EE. Two new plink-BED files are generated from each one of the 6 basic scenarios. Taking the plink-fam files from the 6 basic datasets and executing a short script in R, a new pool of plink-fam files have been

created with the reduced number of populations being the process guided by the random or contagious sampling schema. Then, using plink with option "keep" we have generated the new plink datasets for random and contagious experimental cases (see column "Population Sampling" in Table 5.2A and 5.2B).

[E] From each of the previous datasets we have generated a new version of them by randomly selecting between 1 and 5 individuals per population (unequal sampling).

[F] Finally each of the combinations has been filtered by keeping only those SNPs that are above MAF>0.05 (new subset) and in a separated variant by keeping those SNPs which have low LD (R^2< 0.5). At this point we have generated the 78 datasets as shown in Table 5.2A and 5.2B.

[G] The five algorithms under analysis have been sequentially executed on each of the 78 datasets using a Linux shell script.

[H] Two types of output files coming from the prior massive program execution: MDS, PCA and SPA generate simple matrices where rows are the individuals and the columns are coordinates in two columns (the number of rows-individuals of each case will depend on the specific sampling case). Admixture and SNMF generate matrices where again rows are the individuals and columns are the K=4 inferred ancestry fractions.

[I] Coordinate matrices coming from MDS, PCA and SPA are evaluated through Mantel and Procrustes Test obtaining the correlations against real individuals coordinates. Cross validation and cross entropy procedures have been performed to identify which value of K has the best predictive value and CLUMPP software has been applied for determining the degree of similarity between the ancestry fractions matrices from Admixture and SNMF of the different sampling cases and its corresponding base full sampling case.

6.2 Algorithmic Approach: Performance of PCA, MDS and SPA algorithms for detecting global individual ancestry in a 2D stepping stone and anisotropic model and the impact of migration rate.

The results from applying Mantel and Procrustes tests to the matrices generated by PCA, MDS and SPA compared to the geographic sampling origin of the simulated individuals are shown in Table 6.1A and 6.1B. The p-value for all Mantel and Procrustes tests is 0.001 indicating that our results are statistically significant at an alpha of 0.05. Since the significance is assessed by permutation tests, we determined the p-value by specifying 999 permutations both in Mantel and Procrustes tests.

Table 6.1A: 2D S.Stone - Mantel and Procrustes correlations between MDS, SPA and PCA inferred coordinates and real coordinates (pvalue = $1{\cdot}10^{-3}$)

Demographic		Migr	Population	Num	Individual	Num	Num	plink	MAF	LD	MDS	MDS	SPA	SPA	PCA	PCA
#	Model	rate	Sampling	Pops	Sampling	Inds	SNPs	(Mb)	Filt	Filt	Mantel	Procrust	Mantel	Procrust	Mantel	Procrust
1	2D S.Stone	0.001	Full model	225	Equal	2250	944,093	531.5	No	No	0.9564	0.9848	0.9377	0.9810	0.9654	0.9876
2	2D S.Stone	0.005	Full model	225	Equal	2250	874,491	492.3	No	No	0.9221	0.9705	0.9303	0.9765	0.9447	0.9795
3	2D S.Stone	0.020	Full model	225	Equal	2250	861,222	484.9	No	No	0.7540	0.8990	0.8616	0.9500	0.8585	0.9449
4	2D S.Stone	0.001	Random	75	Equal	750	944,668	177.6	No	No	0.9508	0.9818	0.9274	0.9761	0.9598	0.9849
5	2D S.Stone	0.001	Random	75	Equal	750	328,919	61.8	0.05	No	0.9494	0.9812	0.9469	0.9807	0.9529	0.9823
6	2D S.Stone	0.001	Random	75	Equal	750	440,162	82.8	No	R2=0.5	0.9500	0.9823	0.8957	0.9657	0.9659	0.9873
7	2D S.Stone	0.001	Random	75	Unequal	225	944,668	53.8	No	No	0.8982	0.9528	0.8741	0.9526	0.8972	0.9530
8	2D S.Stone	0.001	Random	75	Unequal	225	327,575	18.7	0.05	No	0.8943	0.9502	0.8851	0.9520	0.8819	0.9448
9	2D S.Stone	0.001	Random	75	Unequal	225	337,558	19.2	No	R2=0.5	0.8996	0.9572	0.8402	0.9468	0.9106	0.9611
10	2D S.Stone	0.001	Contagious	74	Equal	740	944,668	174.8	No	No	0.9586	0.9849	0.9119	0.9715	0.9661	0.9873
11	2D S.Stone	0.001	Contagious	74	Equal	740	328,454	60.8	0.05	No	0.9577	0.9845	0.9524	0.9828	0.9612	0.9854
12	2D S.Stone	0.001	Contagious	74	Equal	740	442,412	81.8	No	R2=0.5	0.9573	0.9852	0.8938	0.9659	0.9705	0.9889
13	2D S.Stone	0.001	Contagious	74	Unequal	211	944,668	50.1	No	No	0.9512	0.9816	0.9042	0.9686	0.9561	0.9834
14	2D S.Stone	0.001	Contagious	74	Unequal	211	325,174	17.2	0.05	No	0.9494	0.9808	0.9455	0.9803	0.9487	0.9806
15	2D S.Stone	0.001	Contagious	74	Unequal	211	334,111	17.7	No	R2=0.5	0.9508	0.9822	0.8791	0.9601	0.9622	0.9858
16	2D S.Stone	0.005	Random	75	Equal	750	874,975	164.5	No	No	0.9109	0.9662	0.9078	0.9674	0.9357	0.9758
17	2D S.Stone	0.005	Random	75	Equal	750	296,178	55.7	0.05	No	0.9037	0.9633	0.9099	0.9660	0.9079	0.9650
18	2D S.Stone	0.005	Random	75	Equal	750	422,053	79.3	No	R2=0.5	0.9244	0.9719	0.9021	0.9679	0.9500	0.9813
19	2D S.Stone	0.005	Random	75	Unequal	225	874,975	49.9	No	No	0.8514	0.9336	0.8245	0.9275	0.8659	0.9408
20	2D S.Stone	0.005	Random	75	Unequal	225	295,573	16.8	0.05	No	0.8435	0.9295	0.8294	0.9251	0.8343	0.9247
21	2D S.Stone	0.005	Random	75	Unequal	225	314,910	17.9	No	R2=0.5	0.8644	0.9404	0.5492	0.6846	0.8825	0.9493
22	2D S.Stone	0.005	Contagious	74	Equal	740	874,975	161.9	No	No	0.9174	0.9687	0.9152	0.9712	0.9404	0.9778
23	2D S.Stone	0.005	Contagious	74	Equal	740	296,679	54.9	0.05	No	0.9099	0.9656	0.9188	0.9695	0.9131	0.9670
24	2D S.Stone	0.005	Contagious	74	Equal	740	421,628	78.0	No	R2=0.5	0.9321	0.9748	0.9119	0.9718	0.9543	0.9831
25	2D S.Stone	0.005	Contagious	74	Unequal	211	874,975	46.4	No	No	0.8928	0.9593	0.5134	0.5590	0.9184	0.9691
26	2D S.Stone	0.005	Contagious	74	Unequal	211	294,351	15.6	0.05	No	0.8839	0.9555	0.8771	0.9527	0.8852	0.9557
27	2D S.Stone	0.005	Contagious	74	Unequal	211	309,956	16.4	No	R2=0.5	0.9112	0.9673	0.8180	0.9274	0.9404	0.9779
28	2D S.Stone	0.020	Random	75	Equal	750	861,222	161.9	No	No	0.6600	0.8504	0.7826	0.9132	0.7845	0.9127
29	2D S.Stone	0.020	Random	75	Equal	750	290,521	54.6	0.05	No	0.6243	0.8309	0.6043	0.8183	0.6566	0.8496
30	2D S.Stone	0.020	Random	75	Equal	750	424,465	79.8	No	R2=0.5	0.7451	0.8939	0.8127	0.9287	0.8472	0.9409
31	2D S.Stone	0.020	Random	75	Unequal	225	861,222	49.1	No	No	0.3789	0.6593	0.0937	0.2527	0.5290	0.7565
32	2D S.Stone	0.020	Random	75	Unequal	225	290,163	16.5	0.05	No	0.3357	0.6214	0.2430	0.4784	0.3846	0.6555
33	2D S.Stone	0.020	Random	75	Unequal	225	314,317	17.9	No	R2=0.5	0.5196	0.7566	0.1058	0.2717	0.6531	0.8306
34	2D S.Stone	0.020	Contagious	74	Equal	740	861,222	159.3	No	No	0.7152	0.8797	0.8243	0.9294	0.8251	0.9301
35	2D S.Stone	0.020	Contagious	74	Equal	740	290,841	53.8	0.05	No	0.6798	0.8621	0.6668	0.8554	0.7056	0.8754
36	2D S.Stone	0.020	Contagious	74	Equal	740	421,952	78.1	No	R2=0.5	0.7892	0.9141	0.8200	0.9313	0.8762	0.9518
37	2D S.Stone	0.020	Contagious	74	Unequal	211	861,222	45.6	No	No	0.4083	0.7171	0.2052	0.4263	0.6122	0.8334
38	2D S.Stone	0.020	Contagious	74	Unequal	211	288,025	15.3	0.05	No	0.3426	0.6717	0.1238	0.3163	0.3988	0.7081
39	2D S.Stone	0.020	Contagious	74	Unequal	211	309,480	16.4	No	R2=0.5	0.5663	0.8048	0.0270	0.1633	0.7230	0.8866

Table 6.1B: Anisotropic - Mantel and Procrustes correlations between MDS, SPA and PCA inferred coordinates and real coordinates (pvalue = $1 \cdot 10^{-3}$)

Demographic		Migr	Population	Num	Individual	Num	Num	plink	MAF	LD	MDS	MDS	SPA	SPA	PCA	PCA
#	Model	rate	Sampling	Pops	Sampling	Inds	SNPs	(Mb)	Filt	Filt	Mantel	Procrust	Mantel	Procrust	Mantel	Procrust
40	Anisotropic	0.001	Full model	125	Equal	1250	503,182	157.5	No	No	0.9318	0.9668	0.9137	0.9677	0.9413	0.9699
41	Anisotropic	0.005	Full model	125	Equal	1250	462,441	144.7	No	No	0.8900	0.9459	0.8769	0.9490	0.9149	0.9571
42	Anisotropic	0.020	Full model	125	Equal	1250	451,116	141.2	No	No	0.7444	0.8794	0.8072	0.9164	0.8388	0.9237
43	Anisotropic	0.001	Random	45	Equal	450	503,390	56.9	No	No	0.8717	0.9138	0.8748	0.9383	0.8363	0.8998
44	Anisotropic	0.001	Random	45	Equal	450	186,436	21.1	0.05	No	0.8692	0.9102	0.8839	0.9379	0.8243	0.8831
45	Anisotropic	0.001	Random	45	Equal	450	213,424	24.1	No	R2=0.5	0.8675	0.9120	0.8572	0.9214	0.8556	0.9199
46	Anisotropic	0.001	Random	45	Unequal	143	503,390	18.1	No	No	0.8462	0.9129	0.6743	0.6447	0.8286	0.9075
47	Anisotropic	0.001	Random	45	Unequal	143	185,538	6.7	0.05	No	0.8406	0.9072	0.8185	0.9250	0.8038	0.8824
48	Anisotropic	0.001	Random	45	Unequal	143	159,830	5.8	No	R2=0.5	0.8280	0.9067	0.8130	0.9221	0.8515	0.9253
49	Anisotropic	0.001	Contagious	45	Equal	450	428,046	48.4	No	No	0.9168	0.9614	0.7132	0.5129	0.9272	0.9643
50	Anisotropic	0.001	Contagious	45	Equal	450	186,857	21.1	0.05	No	0.9123	0.9594	0.8763	0.9494	0.9150	0.9601
51	Anisotropic	0.001	Contagious	45	Equal	450	216,714	24.5	No	R2=0.5	0.9109	0.9609	0.8435	0.9419	0.9331	0.9661
52	Anisotropic	0.001	Contagious	45	Unequal	137	503,390	17.6	No	No	0.8830	0.9484	0.8514	0.9469	0.8827	0.9495
53	Anisotropic	0.001	Contagious	45	Unequal	137	186,885	6.5	0.05	No	0.8747	0.9444	0.8913	0.9543	0.8570	0.9384
54	Anisotropic	0.001	Contagious	45	Unequal	137	158,802	5.6	No	R2=0.5	0.8910	0.9544	0.7868	0.8874	0.9048	0.9583
55	Anisotropic	0.005	Random	45	Equal	450	462,587	52.3	No	No	0.8436	0.9129	0.8871	0.9508	0.8304	0.9095
56	Anisotropic	0.005	Random	45	Equal	450	164,598	18.6	0.05	No	0.8382	0.9097	0.8316	0.9169	0.8122	0.8981
57	Anisotropic	0.005	Random	45	Equal	450	207,715	23.5	No	R2=0.5	0.8533	0.9178	0.8718	0.9443	0.8439	0.9181
58	Anisotropic	0.005	Random	45	Unequal	143	462,587	16.7	No	No	0.7730	0.8781	0.7728	0.9012	0.7625	0.8806
59	Anisotropic	0.005	Random	45	Unequal	143	163,384	5.9	0.05	No	0.7651	0.8726	0.7403	0.8596	0.7356	0.8609
60	Anisotropic	0.005	Random	45	Unequal	143	151,457	5.5	No	R2=0.5	0.7854	0.8832	0.6021	0.7271	0.7869	0.8922
61	Anisotropic	0.005	Contagious	45	Equal	450	462,587	52.3	No	No	0.8732	0.9399	0.8734	0.9473	0.9009	0.9522
62	Anisotropic	0.005	Contagious	45	Equal	450	164,530	18.6	0.05	No	0.8648	0.9358	0.8925	0.9501	0.8691	0.9371
63	Anisotropic	0.005	Contagious	45	Equal	450	209,720	23.7	No	R2=0.5	0.8868	0.9453	0.8625	0.9502	0.9185	0.9594
64	Anisotropic	0.005	Contagious	45	Unequal	137	462,587	16.2	No	No	0.8357	0.9295	0.7838	0.9074	0.8536	0.9380
65	Anisotropic	0.005	Contagious	45	Unequal	137	164,583	5.8	0.05	No	0.8292	0.9262	0.8124	0.9155	0.8317	0.9268
66	Anisotropic	0.005	Contagious	45	Unequal	137	149,677	5.2	No	R2=0.5	0.8544	0.9367	0.7212	0.8821	0.8798	0.9485
67	Anisotropic	0.020	Random	45	Equal	450	451,116	51.0	No	No	0.6408	0.7843	0.5626	0.6721	0.7140	0.8274
68	Anisotropic	0.020	Random	45	Equal	450	160,374	18.1	0.05	No	0.6090	0.7666	0.6091	0.7697	0.6081	0.7657
69	Anisotropic	0.020	Random	45	Equal	450	207,884	23.5	No	R2=0.5	0.7216	0.8355	0.6969	0.8322	0.7616	0.8670
70	Anisotropic	0.020	Random	45	Unequal	143	451,116	16.2	No	No	0.4472	0.6413	0.3142	0.4886	0.4749	0.6413
71	Anisotropic	0.020	Random	45	Unequal	143	158,862	5.7	0.05	No	0.4292	0.6316	0.4633	0.6719	0.4256	0.6280
72	Anisotropic	0.020	Random	45	Unequal	143	148,636	5.4	No	R2=0.5	0.5364	0.7192	0.3389	0.5105	0.5599	0.7045
73	Anisotropic	0.020	Contagious	45	Equal	450	451,116	51.0	No	No	0.6383	0.8275	0.7162	0.8672	0.7484	0.8852
74	Anisotropic	0.020	Contagious	45	Equal	450	160,560	18.1	0.05	No	0.6028	0.8069	0.5938	0.7968	0.6172	0.8157
75	Anisotropic	0.020	Contagious	45	Equal	450	208,853	23.6	No	R2=0.5	0.7202	0.8701	0.6959	0.8615	0.8123	0.9160
76	Anisotropic	0.020	Contagious	45	Unequal	137	451,116	15.8	No	No	0.4688	0.7367	0.2042	0.3951	0.5956	0.8170
77	Anisotropic	0.020	Contagious	45	Unequal	137	160,477	5.6	0.05	No	0.3892	0.6823	0.3284	0.6016	0.4778	0.7462
78	Anisotropic	0.020	Contagious	45	Unequal	137	147,651	5.2	No	R2=0.5	0.5064	0.7600	0.0789	0.2631	0.6586	0.8522

Before performing a more formal statistical assessment of the results we can identify some trends by a simple visual inspection of Table 6.1A and 6.1B: there is a high degree of correlation between the coordinates estimated by the three algorithms and the real data, specially for the six full model used as a reference (highlighted in grey in both tables). The average correlation for these 2D stepping stone and Anisotropic full models with the three algorithms together is 0.93 and 0.91 respectively. Additionally, from the total number of coefficient correlations calculated (78 cases x 3 algorithms x 2 tests = 468) there are 344 correlations (74%) above 0.80 and the rest of the measures (124) showing that low degree of correlation are mainly present (81%) in the high migration rate sections (m=0.02), suggesting in a first approach an inverse relationship between correlation results and migration rate.

Best global correlation mean on the 78 experimental datasets corresponds to PCA with 0.82 and 0.91 for Mantel and Procrustes respectively followed by MDS with 0.78 and 0.89 and finally SPA with 0.72 and 0.83. It is also evident the higher correlation values of Procrustes test compared with Mantel.

In order to identify the best performing algorithm we have applied the one-tail paired Wilcoxon signed-rank test on PCA, MDS and SPA correlations resulting from Mantel and Procrustes tests. The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used when comparing repeated measurements on a single sample to assess whether their population mean ranks differ. Wilcoxon test has been employed fragmenting the results by the two basic demographic models and by the three levels of migration rate and comparing the three algorithms in pairs.

Table 6.2 shows the resulting p-values from all Wilcoxon test rounds: in all the cases PCA is the most robust algorithm followed by MDS.

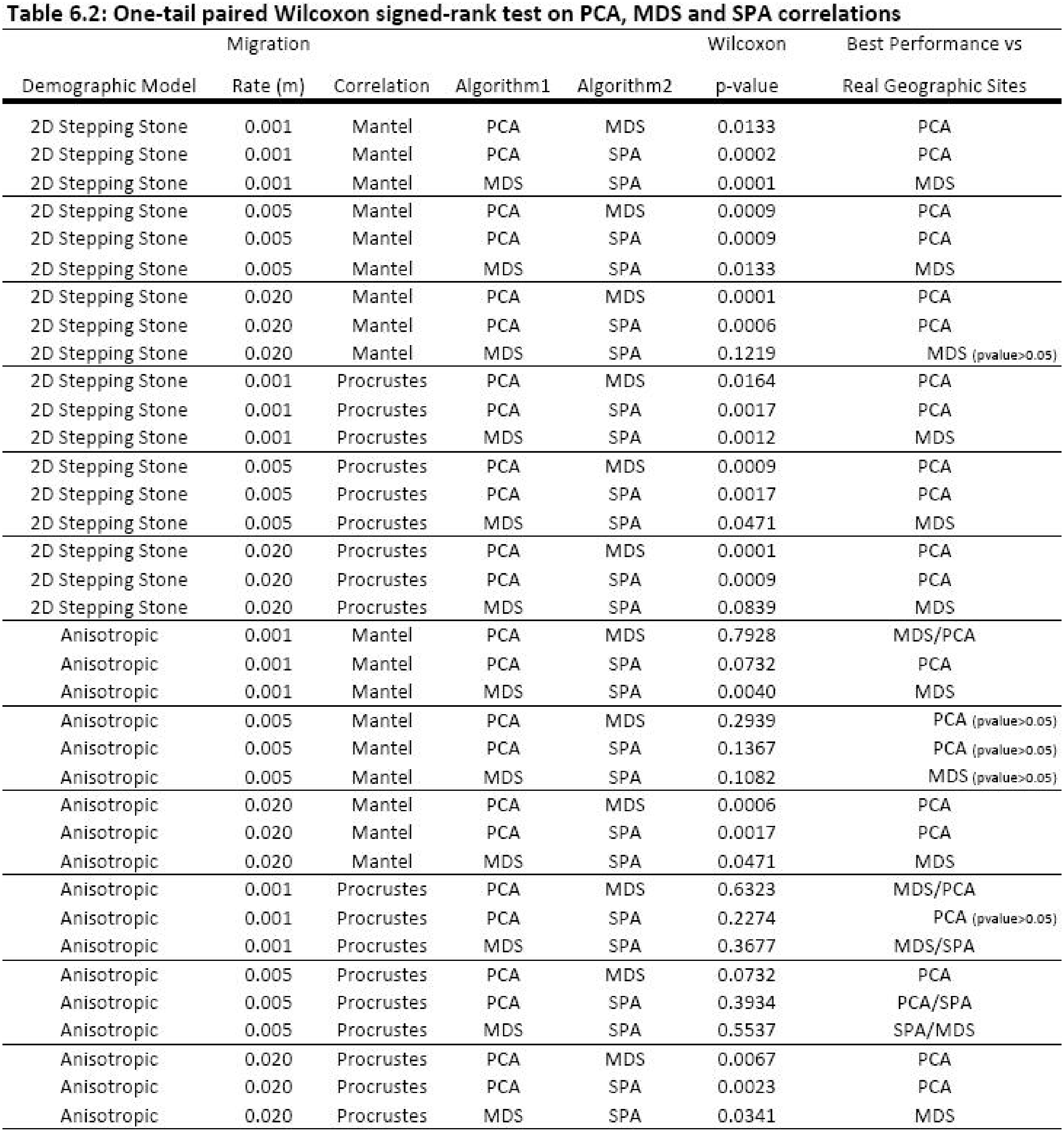


Table 6.2: One-tail paired Wilcoxon signed-rank test on PCA, MDS and SPA correlations

	Migration				Wilcoxon	Best Performance vs
Demographic Model	Rate (m)	Correlation	Algorithm1	Algorithm2	p-value	Real Geographic Sites
2D Stepping Stone	0.001	Mantel	PCA	MDS	0.0133	PCA
2D Stepping Stone	0.001	Mantel	PCA	SPA	0.0002	PCA
2D Stepping Stone	0.001	Mantel	MDS	SPA	0.0001	MDS
2D Stepping Stone	0.005	Mantel	PCA	MDS	0.0009	PCA
2D Stepping Stone	0.005	Mantel	PCA	SPA	0.0009	PCA
2D Stepping Stone	0.005	Mantel	MDS	SPA	0.0133	MDS
2D Stepping Stone	0.020	Mantel	PCA	MDS	0.0001	PCA
2D Stepping Stone	0.020	Mantel	PCA	SPA	0.0006	PCA
2D Stepping Stone	0.020	Mantel	MDS	SPA	0.1219	MDS (pvalue>0.05)
2D Stepping Stone	0.001	Procrustes	PCA	MDS	0.0164	PCA
2D Stepping Stone	0.001	Procrustes	PCA	SPA	0.0017	PCA
2D Stepping Stone	0.001	Procrustes	MDS	SPA	0.0012	MDS
2D Stepping Stone	0.005	Procrustes	PCA	MDS	0.0009	PCA
2D Stepping Stone	0.005	Procrustes	PCA	SPA	0.0017	PCA
2D Stepping Stone	0.005	Procrustes	MDS	SPA	0.0471	MDS
2D Stepping Stone	0.020	Procrustes	PCA	MDS	0.0001	PCA
2D Stepping Stone	0.020	Procrustes	PCA	SPA	0.0009	PCA
2D Stepping Stone	0.020	Procrustes	MDS	SPA	0.0839	MDS
Anisotropic	0.001	Mantel	PCA	MDS	0.7928	MDS/PCA
Anisotropic	0.001	Mantel	PCA	SPA	0.0732	PCA
Anisotropic	0.001	Mantel	MDS	SPA	0.0040	MDS
Anisotropic	0.005	Mantel	PCA	MDS	0.2939	PCA (pvalue>0.05)
Anisotropic	0.005	Mantel	PCA	SPA	0.1367	PCA (pvalue>0.05)
Anisotropic	0.005	Mantel	MDS	SPA	0.1082	MDS (pvalue>0.05)
Anisotropic	0.020	Mantel	PCA	MDS	0.0006	PCA
Anisotropic	0.020	Mantel	PCA	SPA	0.0017	PCA
Anisotropic	0.020	Mantel	MDS	SPA	0.0471	MDS
Anisotropic	0.001	Procrustes	PCA	MDS	0.6323	MDS/PCA
Anisotropic	0.001	Procrustes	PCA	SPA	0.2274	PCA (pvalue>0.05)
Anisotropic	0.001	Procrustes	MDS	SPA	0.3677	MDS/SPA
Anisotropic	0.005	Procrustes	PCA	MDS	0.0732	PCA
Anisotropic	0.005	Procrustes	PCA	SPA	0.3934	PCA/SPA
Anisotropic	0.005	Procrustes	MDS	SPA	0.5537	SPA/MDS
Anisotropic	0.020	Procrustes	PCA	MDS	0.0067	PCA
Anisotropic	0.020	Procrustes	PCA	SPA	0.0023	PCA
Anisotropic	0.020	Procrustes	MDS	SPA	0.0341	MDS

We can visualise by plotting the full set of Mantel and Procrustes correlations to double check this conclusion: Figures 6.2, 6.3, 6.4 and 6.5 display the boxplots comparing the three algorithms in the two demographic models split by the three migration rates:

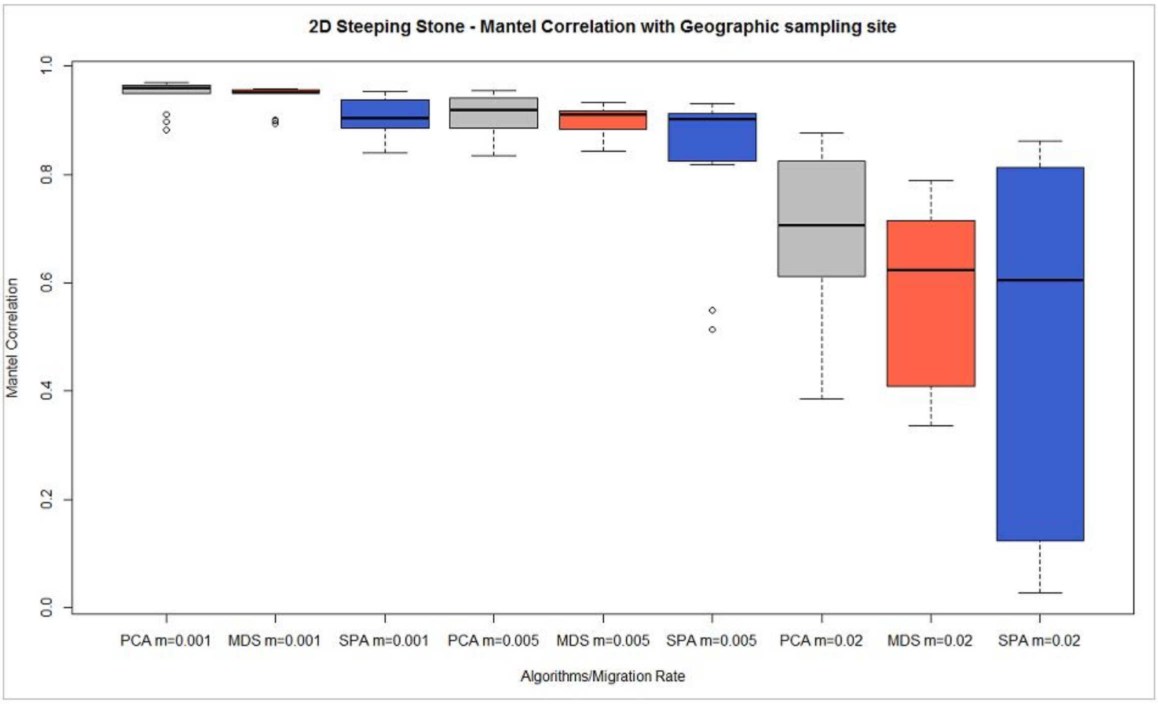


Figure 6.2: Plot with Mantel correlations for 2D Stepping Stone model results from PCA (gray), MDS (red) and SPA (blue) split by migration rates. PCA results show the highest degree of correlation with the geographic sampling origin of the simulated samples.

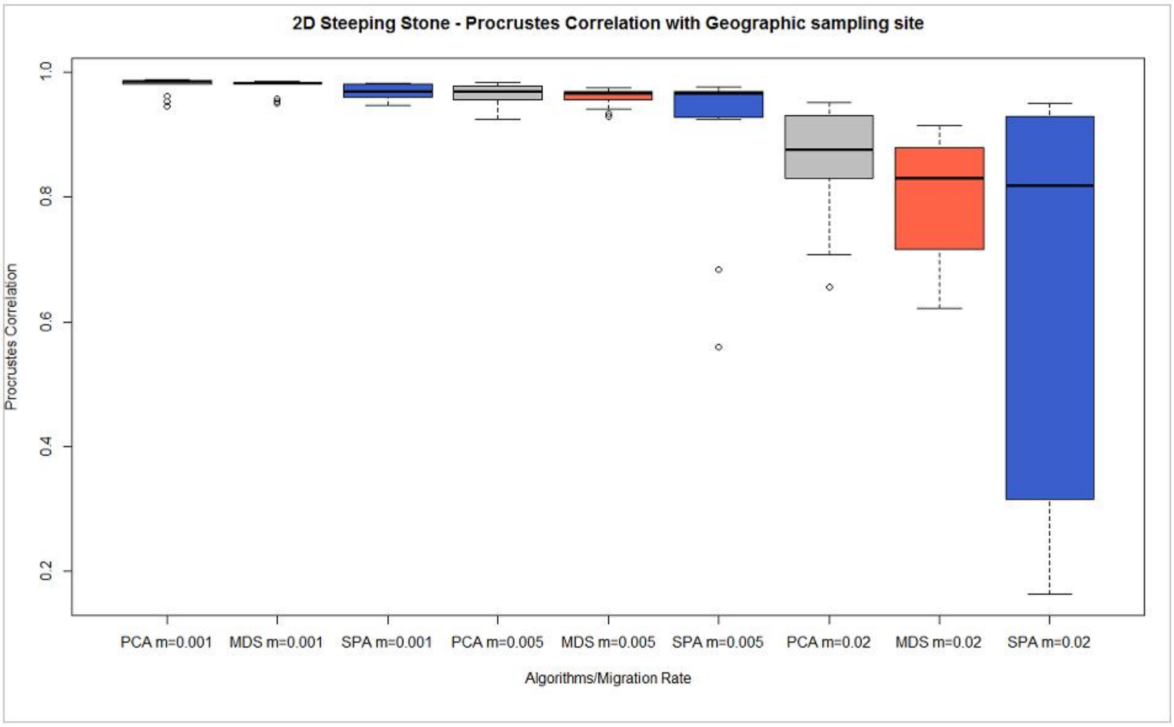


Figure 6.3: Plot with Procrustes correlations for 2D Stepping Stone model results from PCA (gray), MDS (red) and SPA (blue) split by migration rates. PCA results show the highest degree of correlation with the geographic sampling origin of the simulated samples.

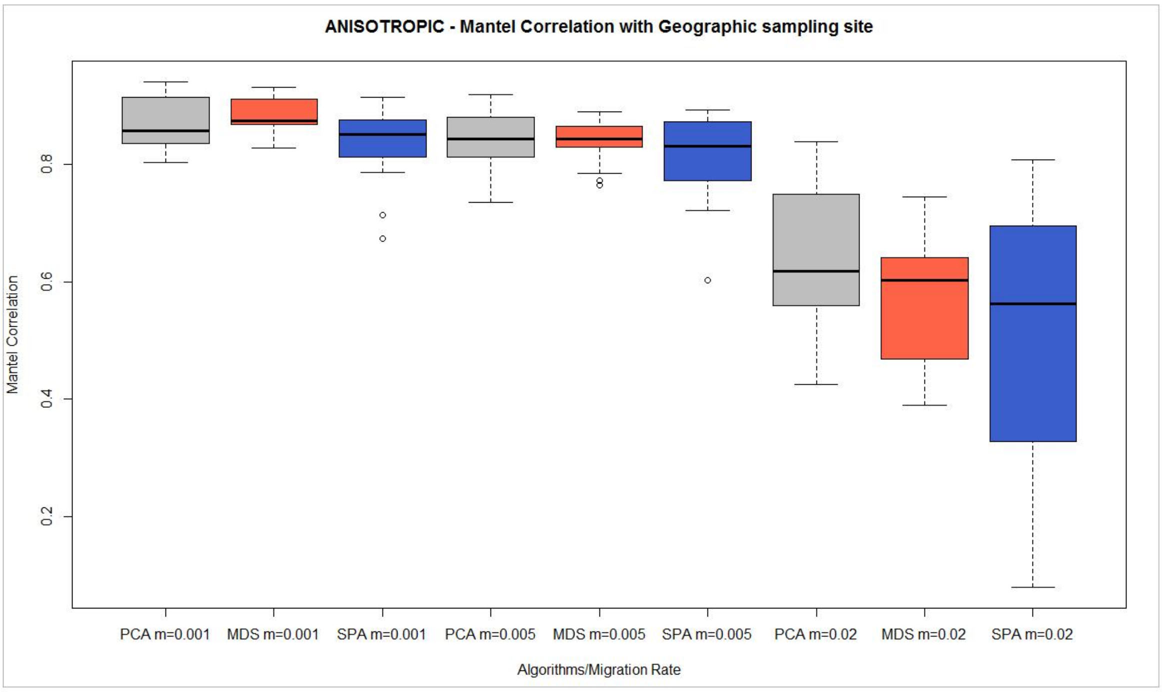


Figure 6.4: Plot with Mantel correlations for Anisotropic model results for 2D Stepping Stone model results from PCA (gray), MDS (red) and SPA (blue) split by migration rates. PCA results show the highest degree of correlation with the geographic sampling origin of the simulated samples.

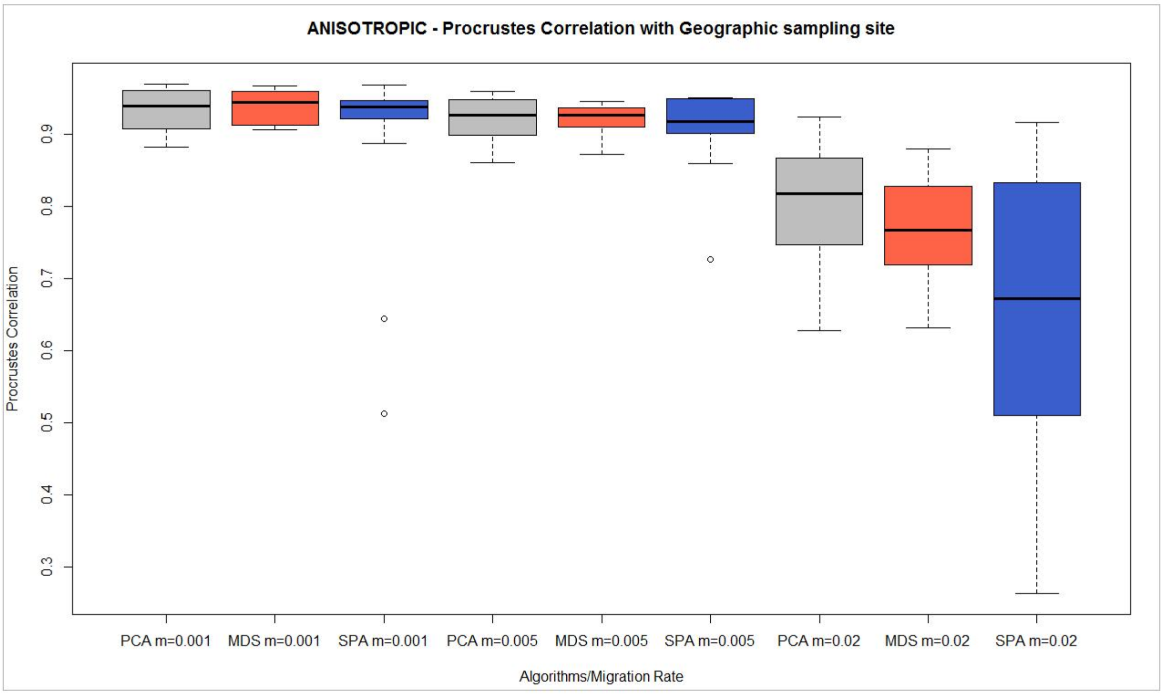


Figure 6.5: Plot with Procrustes correlations for Anisotropic model results for 2D Stepping Stone model results from PCA (gray), MDS (red) and SPA (blue) split by migration rates. PCA results show the highest degree of correlation with the geographic sampling origin of the simulated samples.

All the different boxplot suggest that PCA is the most robust algorithm for estimating the coordinates or geographical location of individuals, closely followed by MDS. They also suggest that the more migration rate, the lower accuracy in the results. To illustrate this relationship, Figure 6.6 displays PCA Mantel correlations in their three levels of migration rate:

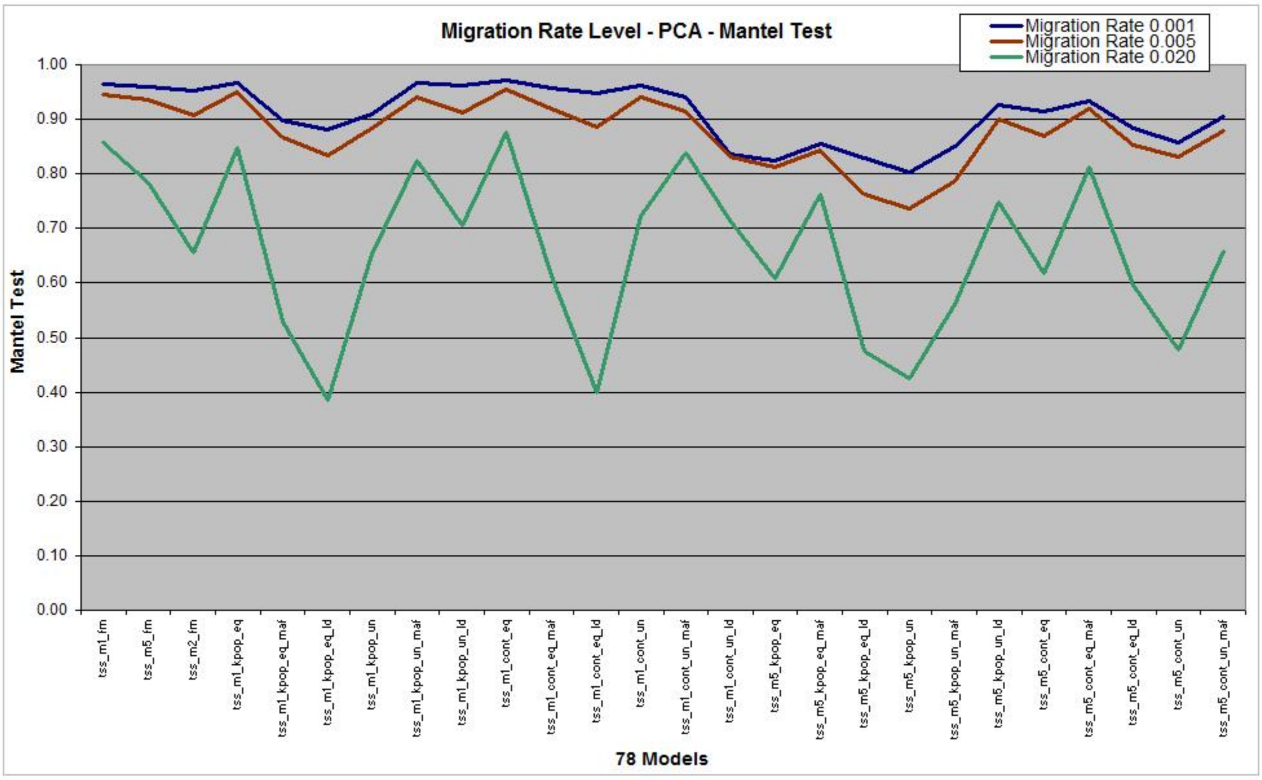
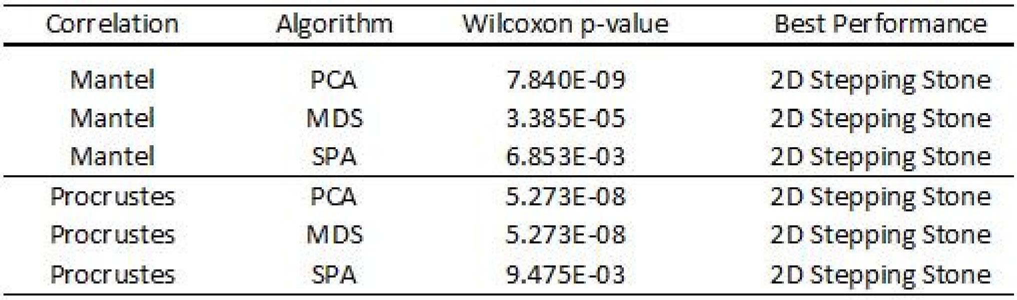


Figure 6.6: PCA Mantel correlations between the 78 datasets and real geographic coordinates split in the three levels of migration rate, becoming evident the decrease of accuracy as the migration increases. Similar behavior has been detected for MDS and SPA algorithms, being the fall due migration more stronger for SPA. Procrustes tests follows the same trend for the three algorithms.

Algorithmic Approach: Impact of the Demographic Model on the Performance of PCA, MDS and SPA

Taking the correlation Mantel and Procrustes results for PCA, MDS and SPA from Table 6.1A and 6.1B and making them independent of migration rate, we can perform Wilcoxon test to evaluate how the demographic design (2D stepping stone or anisotropic) is impacting on the algorithm performance. In this case we are interested in comparing the Mantel correlations for PCA on the 2D stepping stone with PCA on the anisotropic model and also pairing the two measures of MDS and the two measures of SPA. Table 6.3 shows the Wilcoxon test p-values that lead us to conclude that the three algorithms perform more robustly under 2D Stepping Stone scenario.

Table 6.3: One-tail paired Wilcoxon signed-rank test on PCA, MDS and SPA between demographic models: Anisotropic vs 2D Stepping Stone



Correlation	Algorithm	Wilcoxon p-value	Best Performance
Mantel	PCA	7.840E-09	2D Stepping Stone
Mantel	MDS	3.385E-05	2D Stepping Stone
Mantel	SPA	6.853E-03	2D Stepping Stone
Procrustes	PCA	5.273E-08	2D Stepping Stone
Procrustes	MDS	5.273E-08	2D Stepping Stone
Procrustes	SPA	9.475E-03	2D Stepping Stone

Similar results are observed when estimating the correlation between the simulated datasets and the sampling location by means of Mantel test

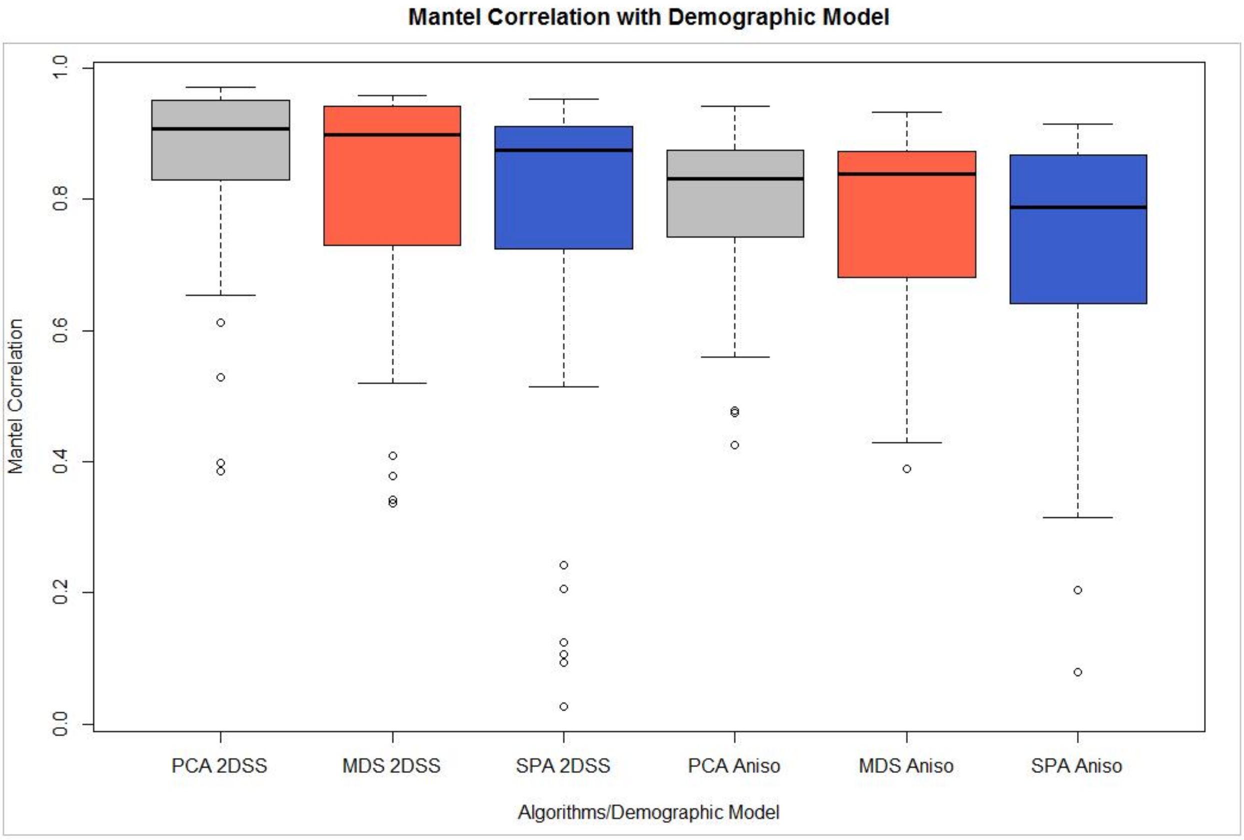


Figure 6.7: Boxplot comparing the three algorithms and their Mantel test correlations with the geographic sampling origin of the simulated samples in the two different scenarios: 2D stepping stone and anisotropic. Each algorithm, PCA (gray), MDS (red) and SPA (blue) performs better under the 2D stepping stone model.

And for Procrustes correlations:

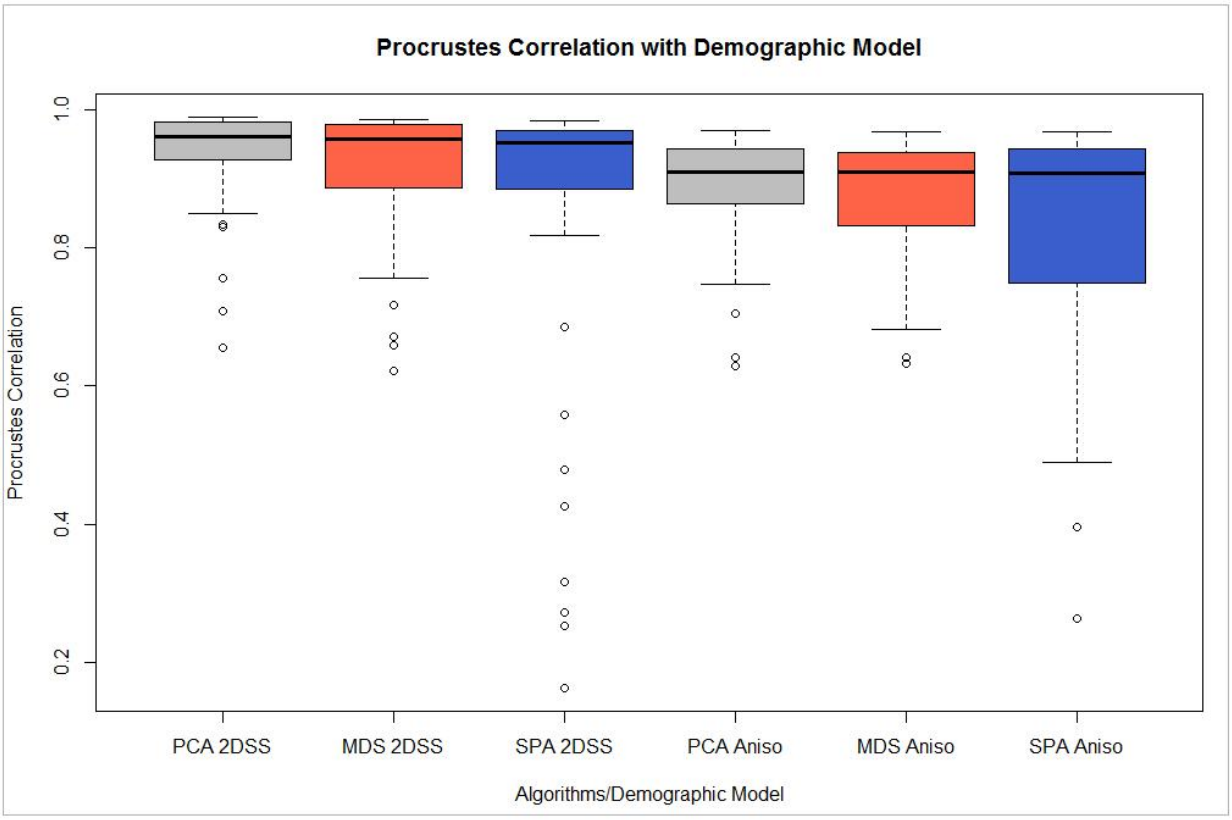


Figure 6.8: Boxplot comparing the three algorithms and their Procrustes test correlations with the geographic sampling origin of the simulated samples in the two different scenarios: 2D stepping stone and anisotropic. Each algorithm performs better under the 2D stepping stone model.

Additionally, and for illustrating purposes, graph (Figure 6.9) shows MDS Mantel correlations comparing by pairs the 39 anisotropic cases with their corresponding 39 2D stepping stone scenarios:

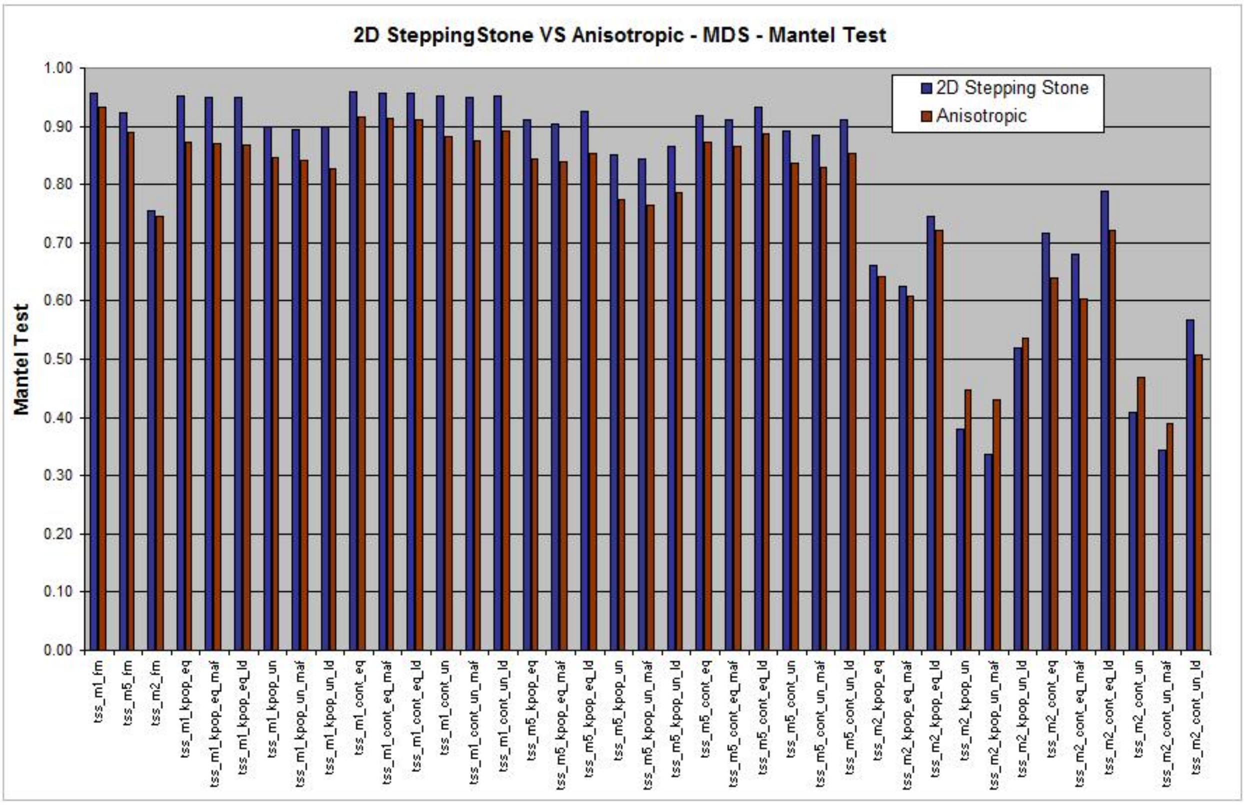


Figure 6.9: Using Mantel Test to estimate the correlation between the coordinates from MDS and the geographic sampling origin of the simulated samples we have obtained these coefficients for the 78 datasets that, paired by the two demographic scenarios, evidence the higher performance of MDS under the 2D stepping stone design.

Algorithmic Approach: Impact of the Population Sampling Method on the Performance of PCA, MDS and SPA

Taking the correlation Mantel and Procrustes results for PCA, MDS and SPA from Table 6.1A and 6.1B and making them independent of all factors except for the population sampling method (Contagious or Random), we can perform Wilcoxon test to evaluate how the way we have selected the populations from the full models is impacting on the algorithm performance. In this case we are interested in comparing the Mantel correlations for PCA on the Contagious Sampling method with PCA on the Random method and also pairing the two measures of MDS and the two measures of SPA.

Table 6.4 show the Wilcoxon test p-values that lead us to conclude that PCA and MDS algorithms perform more robustly under Contagious scenario (according to SPA p-values we can not determine which is the best population sampling method for this algorithm):

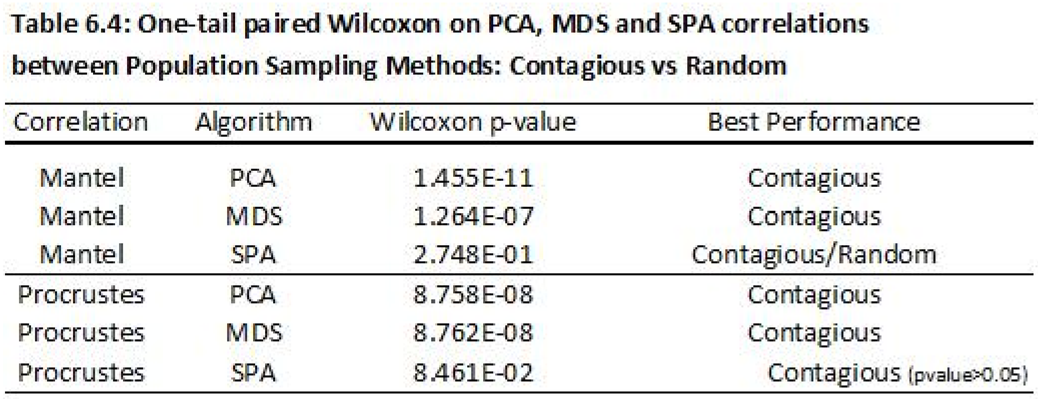


Table 6.4: One-tail paired Wilcoxon on PCA, MDS and SPA correlations between Population Sampling Methods: Contagious vs Random

Correlation	Algorithm	Wilcoxon p-value	Best Performance
Mantel	PCA	1.455E-11	Contagious
Mantel	MDS	1.264E-07	Contagious
Mantel	SPA	2.748E-01	Contagious/Random
Procrustes	PCA	8.758E-08	Contagious
Procrustes	MDS	8.762E-08	Contagious
Procrustes	SPA	8.461E-02	Contagious (pvalue>0.05)

And visually,

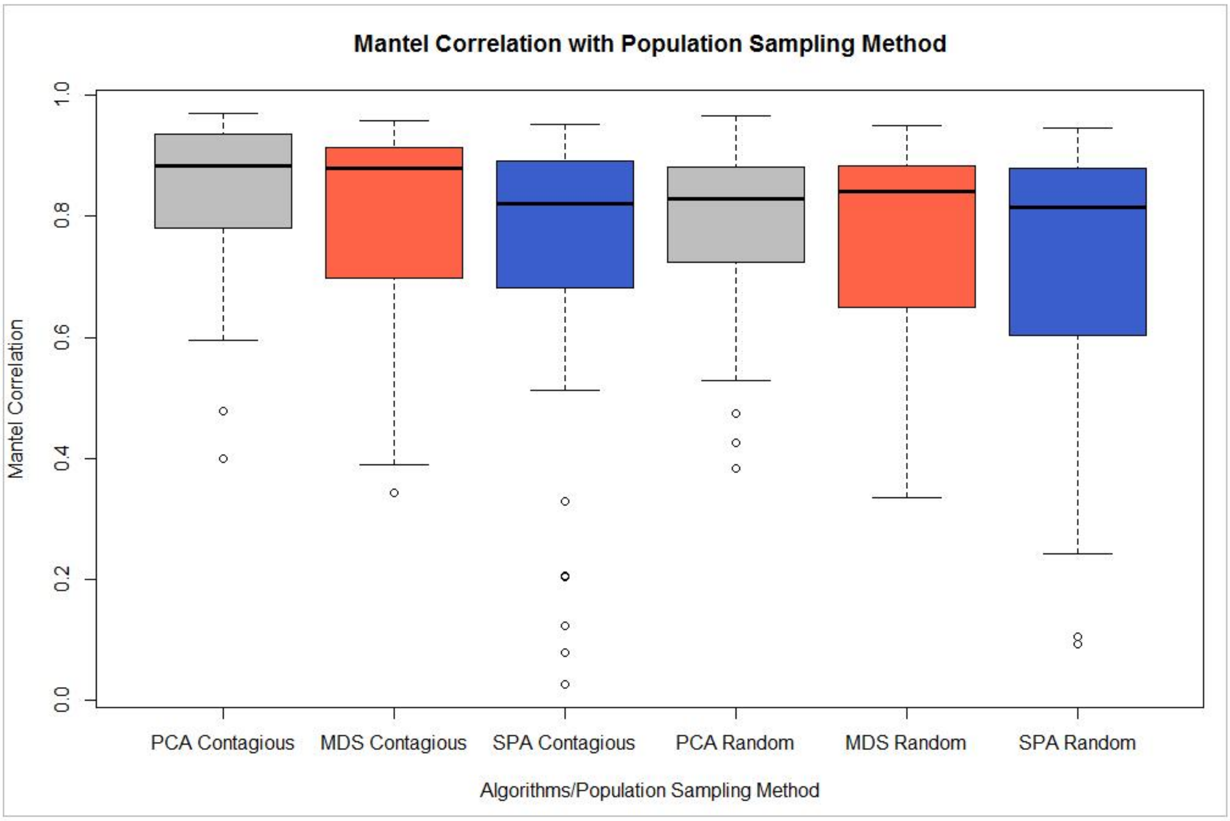


Figure 6.10: Boxplot comparing the Mantel correlations of the three algorithms under the Contagious and Random population sampling methods denoting a slightly stronger degree of correlation for Contagious scenarios when using PCA and MDS algorithms while is not conclusive for SPA algorithm.

And for Procrustes correlations:

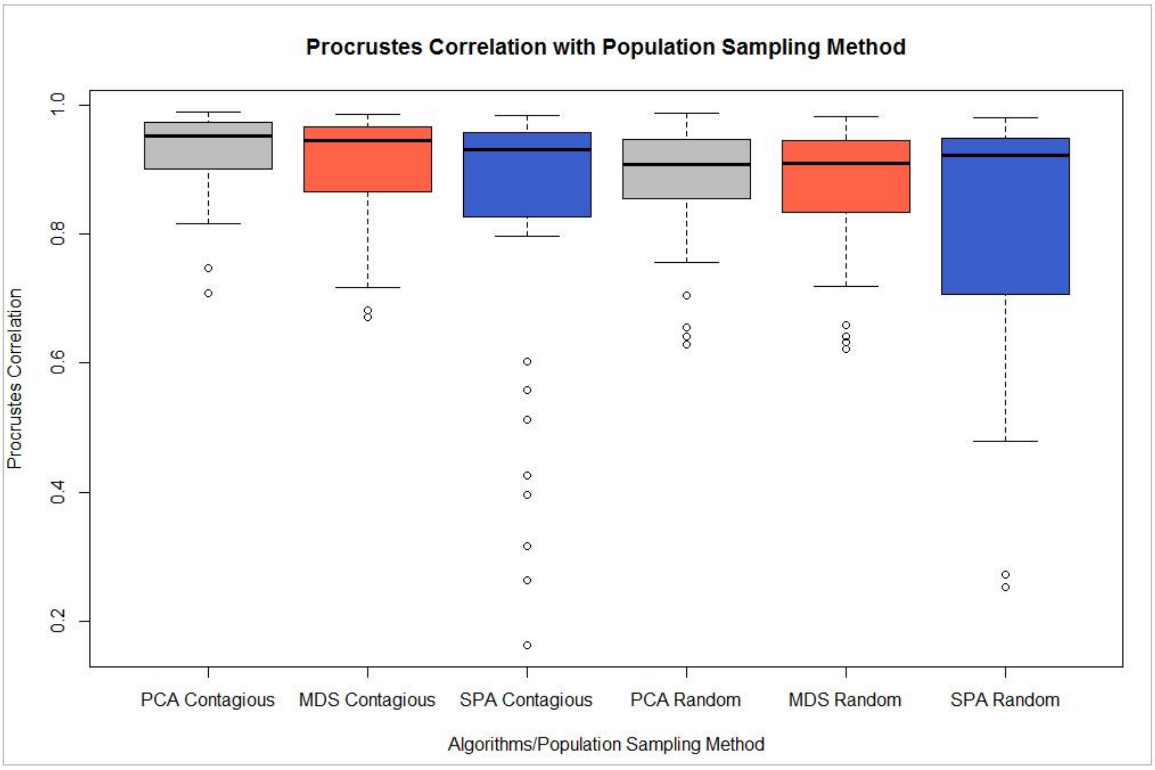


Figure 6.11: Boxplot comparing the Procrustes correlations of the three algorithms under the Contagious and Random population sampling methods denoting a slightly stronger degree of correlation for Contagious scenarios when using PCA and MDS and SPA algorithms.

Figure 6.12 shows PCA Mantel correlations comparing by pairs the 36 contagious cases with their corresponding 36 Random scenarios, making it clear that Contagious values are above Random:

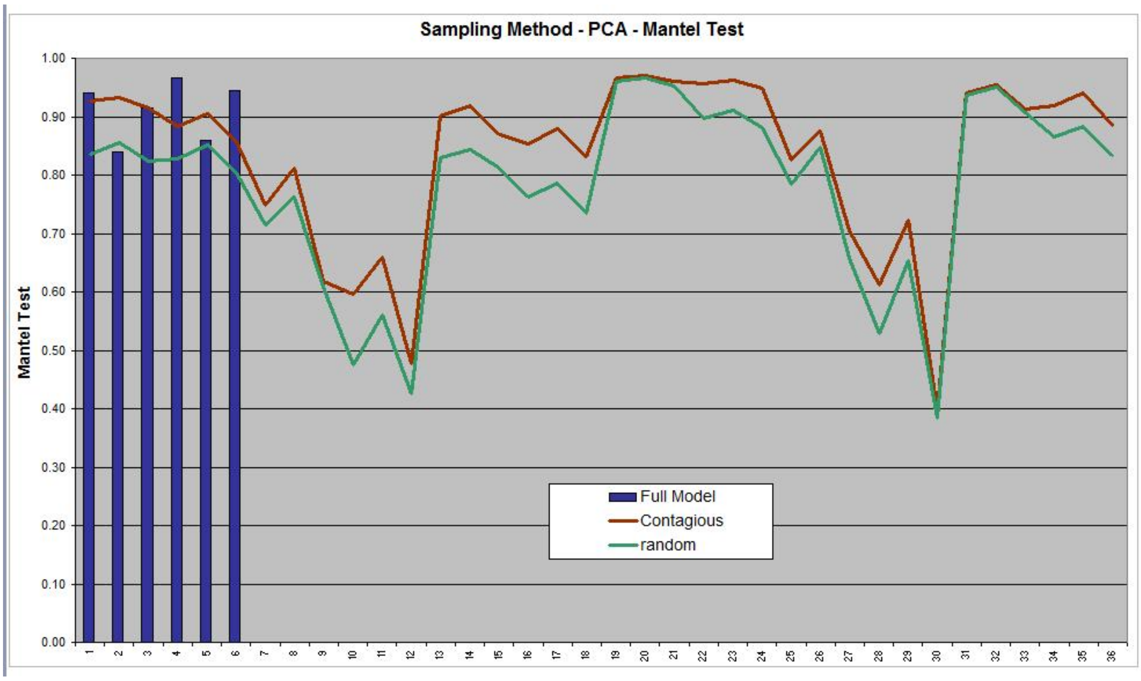


Figure 6.12: Contagious vs Random correlations for PCA (Mantel values): the comparison by pairs of the 36 PCA Mantel correlations for the Contagious models (red) to their equivalent 36 Random (green) denotes the best performance achieved by Contagious population sampling method. Full models are shown on the left (blue) as reference.

Algorithmic Approach: Impact of the Individuals Sampling Method on the Performance of PCA-MDS-SPA

Taking the correlation Mantel and Procrustes results for PCA, MDS and SPA from Table 6.1A and 6.1B and making them independent of all factors except for the individuals sampling method (Equal or Unequal), we can perform Wilcoxon test to evaluate how the way we have selected the individuals within populations is impacting on the algorithm performance. In this case we are interested in comparing the Mantel correlations for PCA on the Equal Sampling method with PCA on the Unequal method and also pairing the two measures of MDS and the two measures of SPA.

Table 6.5 show the Wilcoxon test p-values that lead us to conclude that the three algorithms perform more robustly under Equal scenarios:

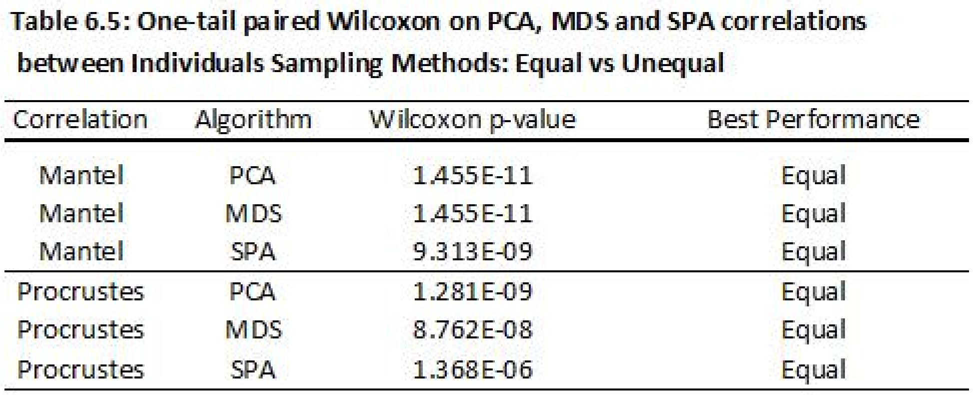


Table 6.5: One-tail paired Wilcoxon on PCA, MDS and SPA correlations between Individuals Sampling Methods: Equal vs Unequal

Correlation	Algorithm	Wilcoxon p-value	Best Performance
Mantel	PCA	1.455E-11	Equal
Mantel	MDS	1.455E-11	Equal
Mantel	SPA	9.313E-09	Equal
Procrustes	PCA	1.281E-09	Equal
Procrustes	MDS	8.762E-08	Equal
Procrustes	SPA	1.368E-06	Equal

Visually:

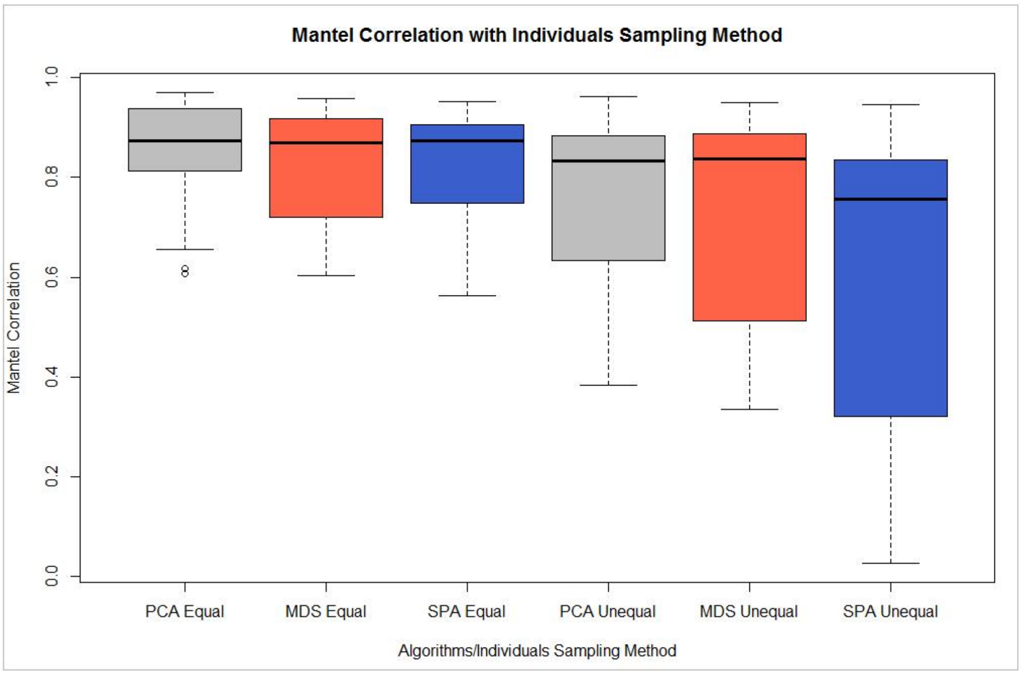


Figure 6.13: The comparison of Mantel correlations split by the two individuals sampling methods (Equal vs Unequal) indicates a higher degree of correlation with the geographic origin of the simulated samples for Equal method and for the three algorithms.

The same for Procrustes test:

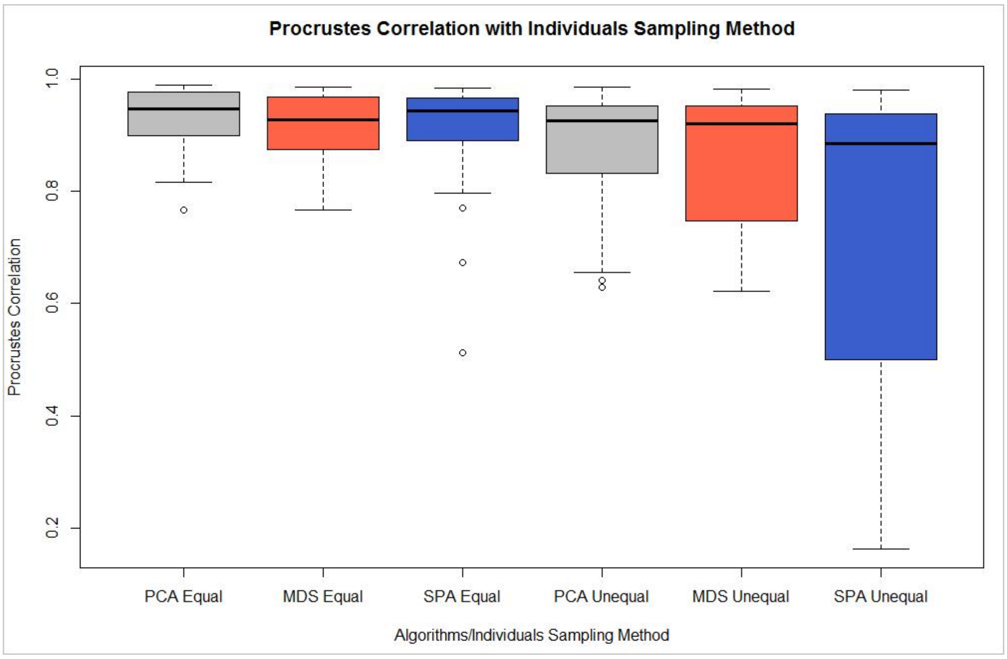


Figure 6.14: Likewise, Procrustes correlations split by the two individuals sampling methods (Equal vs Unequal) also indicates a higher degree of correlation with the geographic origin of the simulated samples for Equal method and for the three algorithms.

And one sample graph showing the behaviour of MDS algorithm under the Equal and Unequal individuals sampling methods (Procrustes):

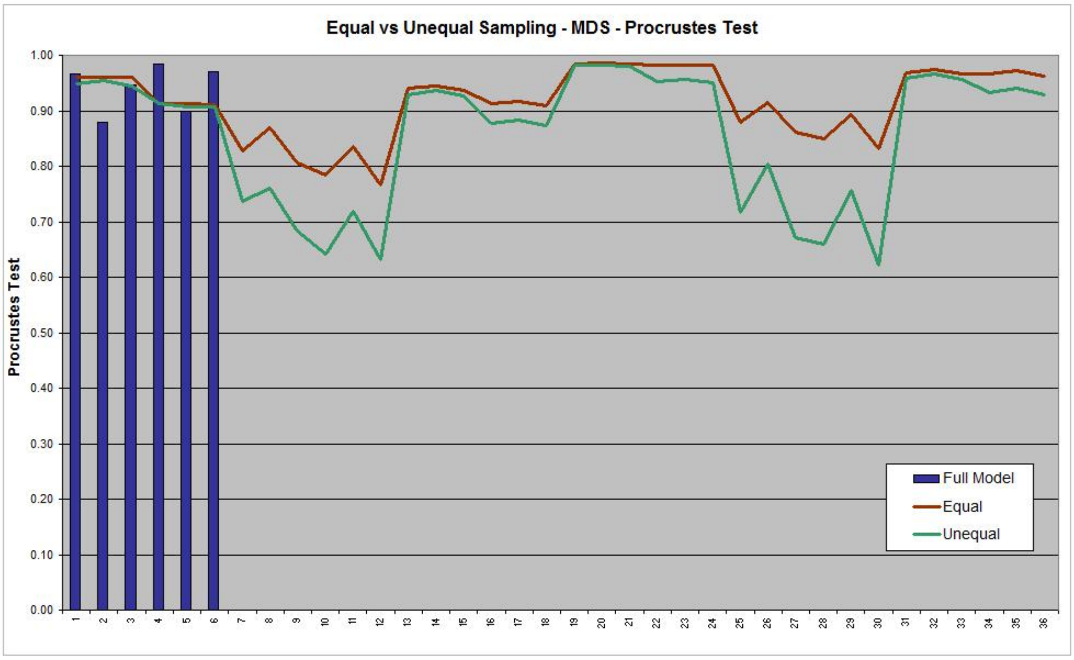


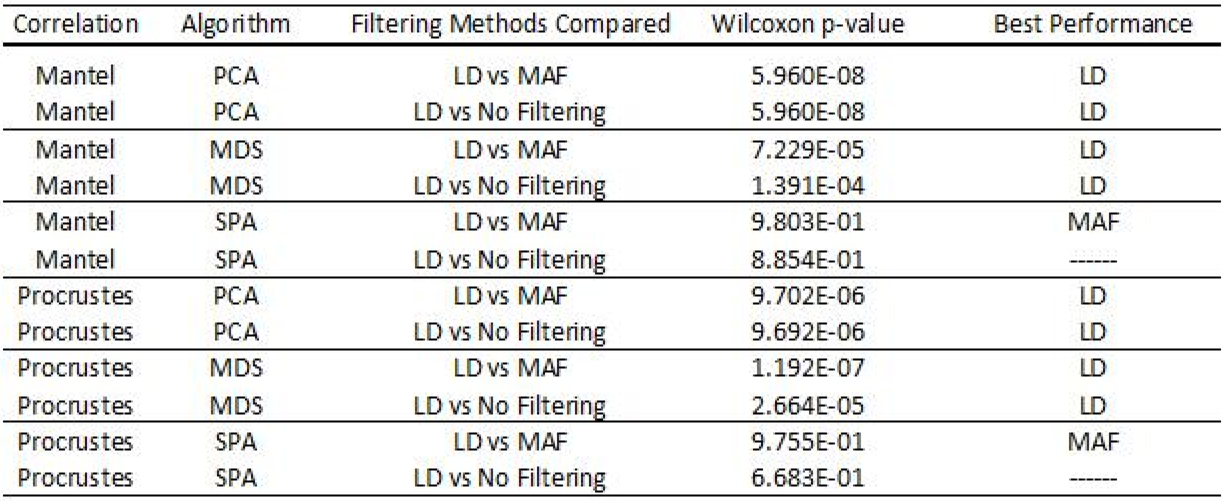
Figure 6.15: Equal vs Unequal Procrustes correlations for MDS: the comparison by pairs of the 36 MDS Procrustes correlations for the Equal sampling models (red) to their equivalent 36 Unequal (green) denotes the best performance achieved by Equal population sampling method in all cases. Full models are shown on the left (blue) as reference.

Algorithmic Approach: Robustness against different levels of data cleaning

We have repeated previously described procedures to assess the impact of filtering methods (MAF and LD) on the three algorithms performance. Now we are comparing three cases: no filtering, MAF<0.05 and LD (R2<0.5).

Table 6.6 show the Wilcoxon test p-values that lead us to conclude that PCA and MDS algorithms perform more robustly under LD data cleaning case, while SPA p-values indicate a better performance for MAF method:

Table 6.6: One-tail paired Wilcoxon on PCA, MDS and SPA correlations between Filtering Methods: LD vs MAF vs No Filtering



Correlation	Algorithm	Filtering Methods Compared	Wilcoxon p-value	Best Performance
Mantel	PCA	LD vs MAF	5.960E-08	LD
Mantel	PCA	LD vs No Filtering	5.960E-08	LD
Mantel	MDS	LD vs MAF	7.229E-05	LD
Mantel	MDS	LD vs No Filtering	1.391E-04	LD
Mantel	SPA	LD vs MAF	9.803E-01	MAF
Mantel	SPA	LD vs No Filtering	8.854E-01	------
Procrustes	PCA	LD vs MAF	9.702E-06	LD
Procrustes	PCA	LD vs No Filtering	9.692E-06	LD
Procrustes	MDS	LD vs MAF	1.192E-07	LD
Procrustes	MDS	LD vs No Filtering	2.664E-05	LD
Procrustes	SPA	LD vs MAF	9.755E-01	MAF
Procrustes	SPA	LD vs No Filtering	6.683E-01	------

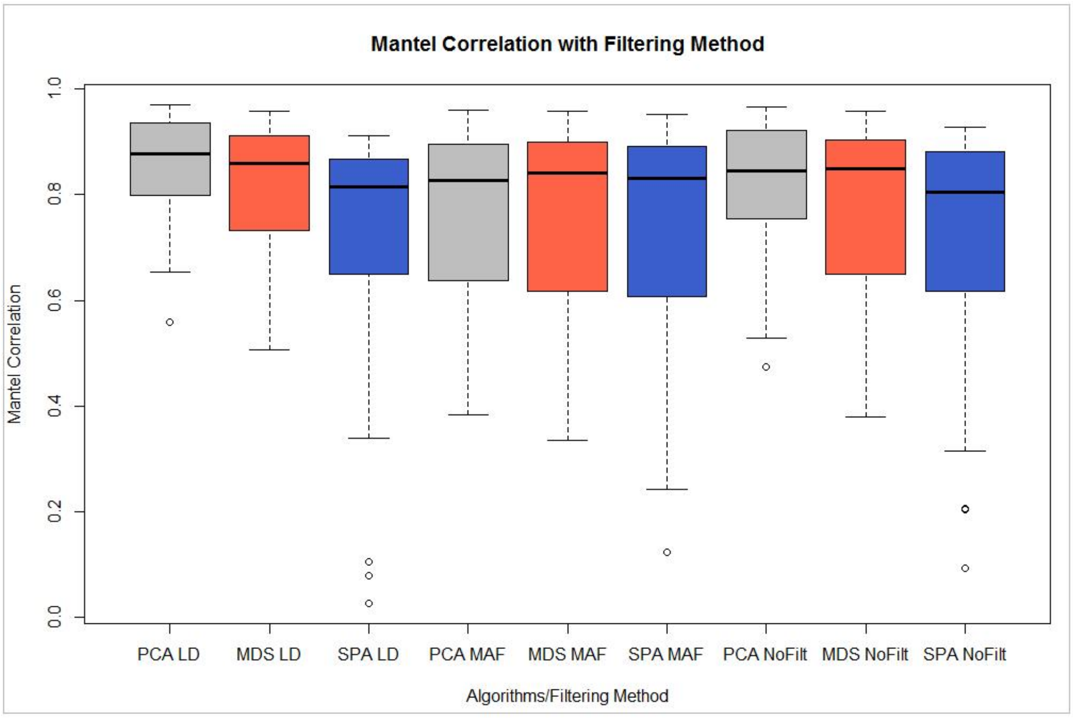


Figure 6.16: Plot with Mantel correlations split by the three cleaning data strategies: LD (left), MAF (center) and No Filtering (right). PCA (gray) and MDS (red) perform with stronger degree of correlation when LD is applied followed by the non filtering case and denoting MAF as the more weak strategy. In contrast, SPA (blue) performs with higher intensity of Mantel correlation when MAF is applied.

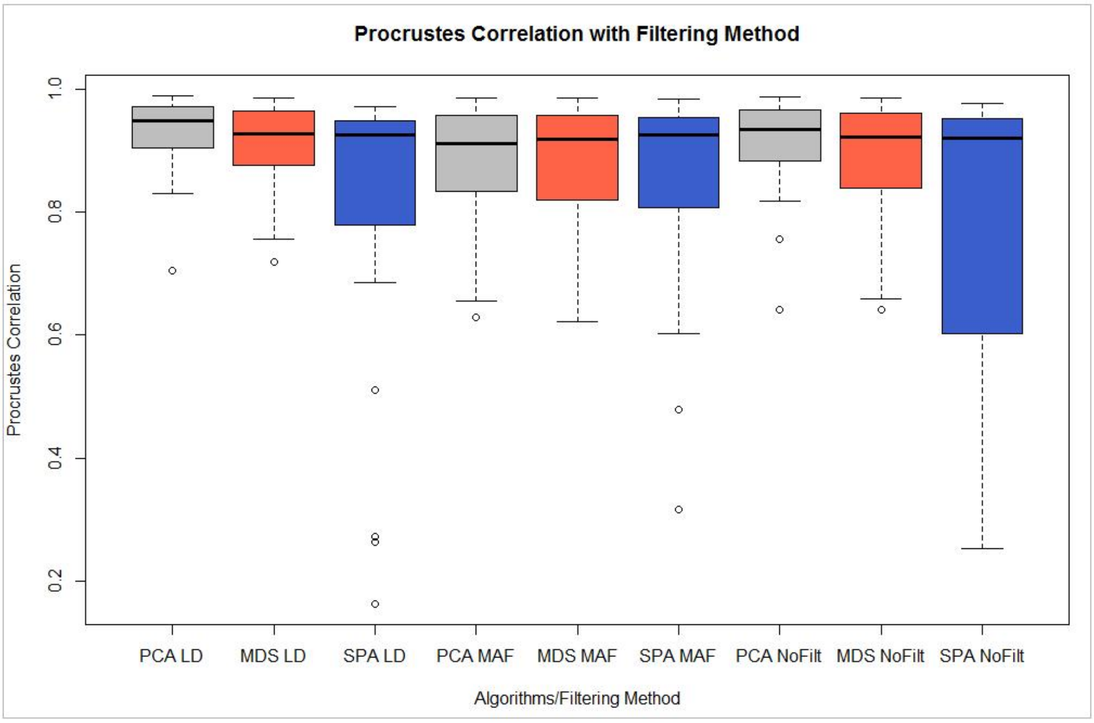


Figure 6.17: Similarly, the Procrustes correlations split by the three cleaning data strategies: LD (left), MAF (center) and No Filtering (right) indicate that PCA (gray) and MDS (red) perform with stronger degree of correlation when LD is applied followed by the non filtering case and denoting MAF as the more weak strategy. In contrast, SPA (blue) performs with higher intensity of Mantel correlation when MAF is applied.

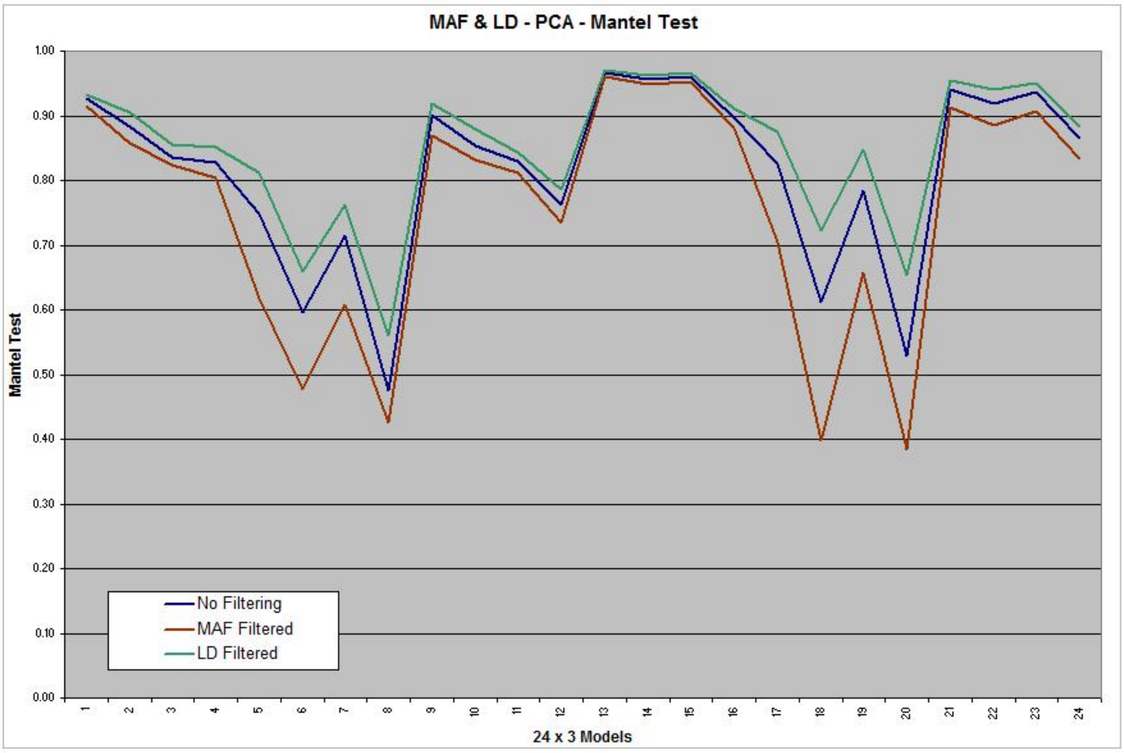


Figure 6.18: The three different cleaning data methods applied to the particular case of PCA (Mantel correlations): the comparison by trios of the 24 PCA Mantel correlations for the LD filtering method (green) , MAF (red) and "No Filtering" (blue) denotes the best performance achieved by LD cleaning method in all cases followed by the "No Filtering" strategy.

6.3 Model Based Approach: Performance of ADMIXTURE and SNMF.

Determining the most predictive K number of ancestry populations

We have sequentially applied the cross validation method (ADMIXTURE) and the cross entropy criterion (SNMF) to identify the best K on the 78 experimental datasets for K={4,5,6,10}. The huge computational resources required for doing this assessment has forced us to narrow the inspection to these particular four values. Additionally, and for having an improved visibility, we have performed a full assessment for K between 1 and 10 using the cross entropy criterion (SNMF) in the particular case of Anisotropic full model with migration rate=0.001 (one of the six reference models):

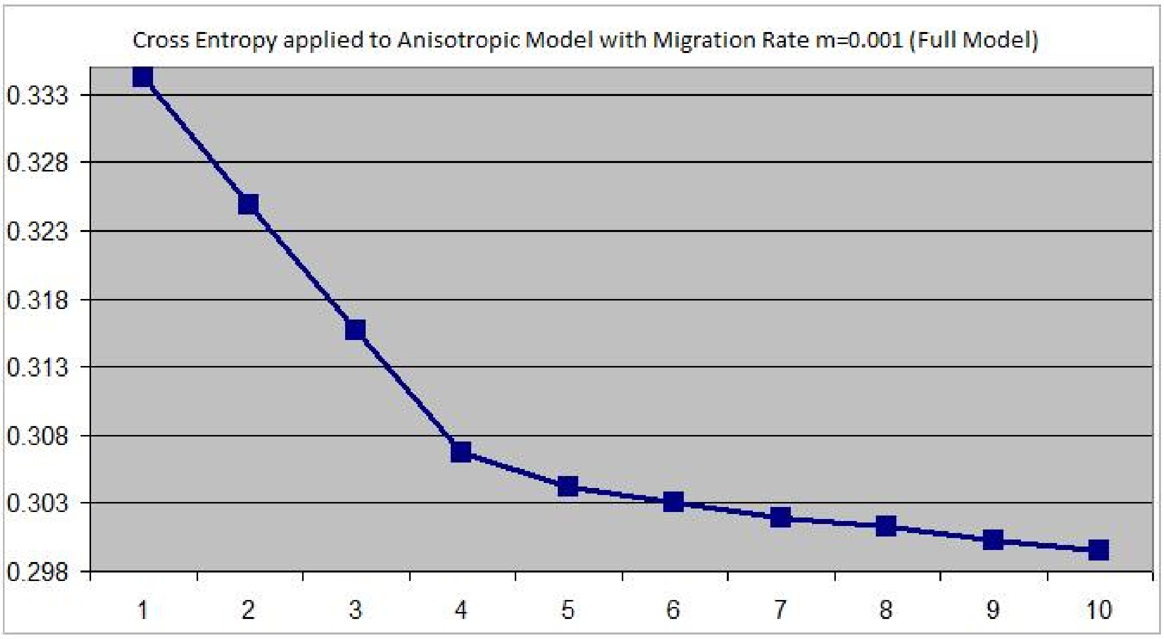


Figure 6.19: Cross entropy error for K between 1 and 10 on Anisotropic referential full model with m=0.001

In this key case, the cross entropy error obtained declines from 0.3343 for K=1 to 0.2995 for K=10 with a dynamic margin of 0.0348 (10% of the maximum value) suggesting a pure asymptotic trend, being K=4 a number of ancestry populations that can be a balanced point in which we minimize the error for ADMIXTURE and SNMF algorithms as well as it makes possible to estimate it in a reasonable amount of time given the high demanding computation time. In fact, the cross entropy error for K=4 is just 20% higher than the error for K=10 (executing ADMIXTURE for K=10 on the 78 experimental datasets need to be measured in “months” rather than hours or days on a server with 64 GB of RAM, 4 processors and Linux 64bits). For these reasons, we have selected K=4 for the 78 runs of ADMIXTURE and SNMF.

Applying CLUMMP test on ADMIXTURE and SNMF

As opposed to the algorithmic approach where PCA, MDS and SPA produce output matrices of N individuals as rows and just 2 columns with the resulting inferred coordinates that can be directly compared with the known real geographical locations, model based algorithms (ADMIXTURE and SNMF) generates output matrices of N individuals and K columns with the estimated ancestry fractions, the elements of each row adding to one. This lack of symmetry is a serious obstacle for the direct comparison of the K ancestry proportions with the geographic origin of the simulated samples. A possible shortcut to coerce matrices symmetry could be based on transforming the K ancestries matrices in an euclidean distance matrix making them comparable to the real geographic origin by using Mantel or Procrustes tests, but this implies some statistical inconsistencies which require further analysis and investigation. As a consequence of this, we have limited our analysis to perform the comparison of the K ancestries matrices from ADMIXTURE and SNMF with their corresponding full sampling cases by means of CLUMPP.

The results from applying CLUMPP test to the matrices generated by ADMIXTURE and SNMF compared to their corresponding full sampling case are shown in Table 6.7. The results also include a third column result showing the CLUMPP correlation results on a paired comparison between ADMIXTURE and SNMF in every experimental dataset.

We can identify some trends by a simple visual inspection of Table 6.7: there is a relatively strong (ADMIXTURE) or high (SNMF) degree of correlation between the K ancestry proportions estimated by the two algorithms and the output matrices for the basic scenarios. The average CLUMPP similarity level (G') for all the ADMIXTURE and SNMF runs are 0.80 and 0.87 respectively, suggesting a better performance of SNMF. The paired comparison ADMIXTURE vs SNMF shows an average similarity of 0.80. Obviously the CLUMPP result for the six full models (highlighted in gray) is equal to 1 since we are comparing their output with themselves. Additionally, from the total number of similarity coefficients G' (72 cases x 2 algorithms = 144) there are 102 correlations (71%) above 0.80 and the rest of the measures (42) showing that low degree of similarity are mainly present (79%) in the high migration rate level (m=0.02), suggesting in a first approach an inverse relationship between correlation results and migration rate.

Table 6.7: CLUMPP correlations between ADMIXTURE and SNMF

Demographic		Migr	Population	Num	Individual	Num	Num	plink	MAF	LD	CLUMPP correlations (G')		
#	Model	rate	Sampling	Pops	Sampling	Inds	SNPs	(Mb)	Filt	Filt	ADMIXTURE	SNMF	ADMIX vs SNMF
1	2D S.Stone	0.001	Full model	225	Equal	2250	944,093	531.5	No	No	1.0000	1.0000	0.9415
2	2D S.Stone	0.005	Full model	225	Equal	2250	874,491	492.3	No	No	1.0000	1.0000	0.8896
3	2D S.Stone	0.020	Full model	225	Equal	2250	861,222	484.9	No	No	1.0000	1.0000	0.8051
4	2D S.Stone	0.001	Random	75	Equal	750	944,668	177.6	No	No	0.9440	0.9506	0.9344
5	2D S.Stone	0.001	Random	75	Equal	750	328,919	61.8	0.05	No	0.9337	0.9516	0.9680
6	2D S.Stone	0.001	Random	75	Equal	750	440,162	82.8	No	R2=0.5	0.8364	0.9308	0.8454
7	2D S.Stone	0.001	Random	75	Unequal	225	944,668	53.8	No	No	0.8481	0.8698	0.9180
8	2D S.Stone	0.001	Random	75	Unequal	225	327,575	18.7	0.05	No	0.8545	0.8700	0.9643
9	2D S.Stone	0.001	Random	75	Unequal	225	337,558	19.2	No	R2=0.5	0.8002	0.6322	0.5334
10	2D S.Stone	0.001	Contagious	74	Equal	740	944,668	174.8	No	No	0.9714	0.9753	0.9339
11	2D S.Stone	0.001	Contagious	74	Equal	740	328,454	60.8	0.05	No	0.9421	0.9732	0.9737
12	2D S.Stone	0.001	Contagious	74	Equal	740	442,412	81.8	No	R2=0.5	0.9036	0.9575	0.8844
13	2D S.Stone	0.001	Contagious	74	Unequal	211	944,668	50.1	No	No	0.9426	0.9497	0.9197
14	2D S.Stone	0.001	Contagious	74	Unequal	211	325,174	17.2	0.05	No	0.9339	0.9522	0.9715
15	2D S.Stone	0.001	Contagious	74	Unequal	211	334,111	17.7	No	R2=0.5	0.4827	0.9341	0.4908
16	2D S.Stone	0.005	Random	75	Equal	750	874,975	164.5	No	No	0.9393	0.9478	0.8788
17	2D S.Stone	0.005	Random	75	Equal	750	296,178	55.7	0.05	No	0.8857	0.9490	0.9661
18	2D S.Stone	0.005	Random	75	Equal	750	422,053	79.3	No	R2=0.5	0.8438	0.9150	0.8234
19	2D S.Stone	0.005	Random	75	Unequal	225	874,975	49.9	No	No	0.8168	0.6698	0.5742
20	2D S.Stone	0.005	Random	75	Unequal	225	295,573	16.8	0.05	No	0.6314	0.8696	0.6515
21	2D S.Stone	0.005	Random	75	Unequal	225	314,910	17.9	No	R2=0.5	0.6877	0.8289	0.7524
22	2D S.Stone	0.005	Contagious	74	Equal	740	874,975	161.9	No	No	0.9487	0.9570	0.8743
23	2D S.Stone	0.005	Contagious	74	Equal	740	296,679	54.9	0.05	No	0.8745	0.9515	0.9672
24	2D S.Stone	0.005	Contagious	74	Equal	740	421,628	78.0	No	R2=0.5	0.8953	0.9313	0.8472
25	2D S.Stone	0.005	Contagious	74	Unequal	211	874,975	46.4	No	No	0.8726	0.9149	0.8305
26	2D S.Stone	0.005	Contagious	74	Unequal	211	294,351	15.6	0.05	No	0.8684	0.9233	0.9423
27	2D S.Stone	0.005	Contagious	74	Unequal	211	309,956	16.4	No	R2=0.5	0.7870	0.8811	0.7784
28	2D S.Stone	0.020	Random	75	Equal	750	861,222	161.9	No	No	0.8655	0.8888	0.7623
29	2D S.Stone	0.020	Random	75	Equal	750	290,521	54.6	0.05	No	0.7659	0.8839	0.9404
30	2D S.Stone	0.020	Random	75	Equal	750	424,465	79.8	No	R2=0.5	0.7632	0.8561	0.6951
31	2D S.Stone	0.020	Random	75	Unequal	225	861,222	49.1	No	No	0.5944	0.7428	0.6468
32	2D S.Stone	0.020	Random	75	Unequal	225	290,163	16.5	0.05	No	0.6459	0.7241	0.7875
33	2D S.Stone	0.020	Random	75	Unequal	225	314,317	17.9	No	R2=0.5	0.3416	0.7051	0.3664
34	2D S.Stone	0.020	Contagious	74	Equal	740	861,222	159.3	No	No	0.8935	0.9080	0.7775
35	2D S.Stone	0.020	Contagious	74	Equal	740	290,841	53.8	0.05	No	0.7712	0.8668	0.8935
36	2D S.Stone	0.020	Contagious	74	Equal	740	421,952	78.1	No	R2=0.5	0.8106	0.8670	0.7398
37	2D S.Stone	0.020	Contagious	74	Unequal	211	861,222	45.6	No	No	0.5392	0.7701	0.5471
38	2D S.Stone	0.020	Contagious	74	Unequal	211	288,025	15.3	0.05	No	0.6434	0.7009	0.6749
39	2D S.Stone	0.020	Contagious	74	Unequal	211	309,480	16.4	No	R2=0.5	0.2980	0.7252	0.3175
40	Anisotropic	0.001	Full model	125	Equal	1250	503,182	157.5	No	No	1.0000	1.0000	0.9481
41	Anisotropic	0.005	Full model	125	Equal	1250	462,441	144.7	No	No	1.0000	1.0000	0.8938
42	Anisotropic	0.020	Full model	125	Equal	1250	451,116	141.2	No	No	1.0000	1.0000	0.7910
43	Anisotropic	0.001	Random	45	Equal	450	503,390	56.9	No	No	0.9453	0.9489	0.9334
44	Anisotropic	0.001	Random	45	Equal	450	186,436	21.1	0.05	No	0.9240	0.9367	0.9616
45	Anisotropic	0.001	Random	45	Equal	450	213,424	24.1	No	R2=0.5	0.9053	0.9443	0.8958
46	Anisotropic	0.001	Random	45	Unequal	143	503,390	18.1	No	No	0.8999	0.9099	0.9322
47	Anisotropic	0.001	Random	45	Unequal	143	185,538	6.7	0.05	No	0.9217	0.9036	0.9543
48	Anisotropic	0.001	Random	45	Unequal	143	159,830	5.8	No	R2=0.5	0.7974	0.8983	0.8497
49	Anisotropic	0.001	Contagious	45	Equal	450	428,046	48.4	No	No	0.9629	0.9682	0.9409
50	Anisotropic	0.001	Contagious	45	Equal	450	186,857	21.1	0.05	No	0.9574	0.9698	0.9715
51	Anisotropic	0.001	Contagious	45	Equal	450	216,714	24.5	No	R2=0.5	0.8904	0.9449	0.8897
52	Anisotropic	0.001	Contagious	45	Unequal	137	503,390	17.6	No	No	0.9304	0.9433	0.9324
53	Anisotropic	0.001	Contagious	45	Unequal	137	186,885	6.5	0.05	No	0.9424	0.9445	0.9663
54	Anisotropic	0.001	Contagious	45	Unequal	137	158,802	5.6	No	R2=0.5	0.8324	0.9208	0.8551
55	Anisotropic	0.005	Random	45	Equal	450	462,587	52.3	No	No	0.9361	0.9405	0.8778
56	Anisotropic	0.005	Random	45	Equal	450	164,598	18.6	0.05	No	0.8826	0.9328	0.9535
57	Anisotropic	0.005	Random	45	Equal	450	207,715	23.5	No	R2=0.5	0.8906	0.9187	0.8547
58	Anisotropic	0.005	Random	45	Unequal	143	462,587	16.7	No	No	0.8406	0.9068	0.8393
59	Anisotropic	0.005	Random	45	Unequal	143	163,384	5.9	0.05	No	0.8833	0.8999	0.9308
60	Anisotropic	0.005	Random	45	Unequal	143	151,457	5.5	No	R2=0.5	0.7624	0.8901	0.7869
61	Anisotropic	0.005	Contagious	45	Equal	450	462,587	52.3	No	No	0.9447	0.9532	0.8798
62	Anisotropic	0.005	Contagious	45	Equal	450	164,530	18.6	0.05	No	0.9048	0.9566	0.9592
63	Anisotropic	0.005	Contagious	45	Equal	450	209,720	23.7	No	R2=0.5	0.8765	0.9183	0.8474
64	Anisotropic	0.005	Contagious	45	Unequal	137	462,587	16.2	No	No	0.8907	0.9020	0.8686
65	Anisotropic	0.005	Contagious	45	Unequal	137	164,583	5.8	0.05	No	0.9086	0.9121	0.9371
66	Anisotropic	0.005	Contagious	45	Unequal	137	149,677	5.2	No	R2=0.5	0.7864	0.8748	0.7715
67	Anisotropic	0.020	Random	45	Equal	450	451,116	51.0	No	No	0.8537	0.7754	0.6241
68	Anisotropic	0.020	Random	45	Equal	450	160,374	18.1	0.05	No	0.7313	0.8268	0.8624
69	Anisotropic	0.020	Random	45	Equal	450	207,884	23.5	No	R2=0.5	0.7748	0.8380	0.7237
70	Anisotropic	0.020	Random	45	Unequal	143	451,116	16.2	No	No	0.5565	0.7031	0.5410
71	Anisotropic	0.020	Random	45	Unequal	143	158,862	5.7	0.05	No	0.6036	0.7495	0.7076
72	Anisotropic	0.020	Random	45	Unequal	143	148,636	5.4	No	R2=0.5	0.2921	0.7618	0.3113
73	Anisotropic	0.020	Contagious	45	Equal	450	451,116	51.0	No	No	0.8716	0.7571	0.5986
74	Anisotropic	0.020	Contagious	45	Equal	450	160,560	18.1	0.05	No	0.7556	0.8325	0.8903
75	Anisotropic	0.020	Contagious	45	Equal	450	208,853	23.6	No	R2=0.5	0.7843	0.8358	0.7191
76	Anisotropic	0.020	Contagious	45	Unequal	137	451,116	15.8	No	No	0.5371	0.6728	0.5159
77	Anisotropic	0.020	Contagious	45	Unequal	137	160,477	5.6	0.05	No	0.6079	0.6910	0.6306
78	Anisotropic	0.020	Contagious	45	Unequal	137	147,651	5.2	No	R2=0.5	0.3689	0.7480	0.3589

In order to identify the best performing algorithm we have applied the one-tail paired Wilcoxon signed-rank test on ADMIXTURE and SNMF similarity coefficients resulting from CLUMPP tests. Wilcoxon test has been employed fragmenting the results by the two basic demographic models and by the three levels of migration rate and comparing the two algorithms in pairs.

Table 6.8 shows the resulting p-values from all Wilcoxon test rounds: in all the cases SNMF is the most robust algorithm. In the second exercise showed in Table 6.8 the resulting Wilcoxon p-values are not conclusive comparing Anisotropic vs 2D Stepping Stone in either of the two algorithms.

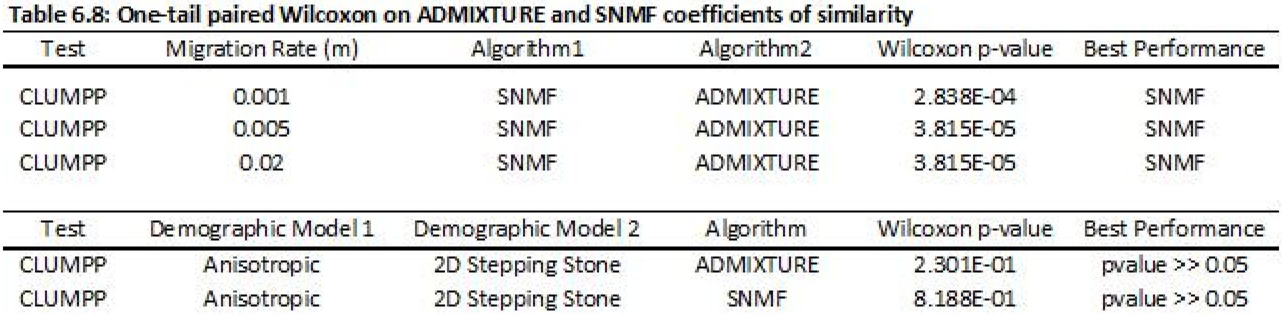


Table 6.8: One-tail paired Wilcoxon on ADMIXTURE and SNMF coefficients of similarity

Test	Migration Rate (m)	Algorithm1	Algorithm2	Wilcoxon p-value	Best Performance
CLUMPP	0.001	SNMF	ADMIXTURE	2.838E-04	SNMF
CLUMPP	0.005	SNMF	ADMIXTURE	3.815E-05	SNMF
CLUMPP	0.02	SNMF	ADMIXTURE	3.815E-05	SNMF

Test	Demographic Model 1	Demographic Model 2	Algorithm	Wilcoxon p-value	Best Performance
CLUMPP	Anisotropic	2D Stepping Stone	ADMIXTURE	2.301E-01	pvalue >> 0.05
CLUMPP	Anisotropic	2D Stepping Stone	SNMF	8.188E-01	pvalue >> 0.05

We can visualise by plotting the CLUMPP correlations to double check these conclusions: Figures 6.20 and 6.21 display the boxplots comparing the two algorithms split by the three migration rates and split by the two demographic models respectively:

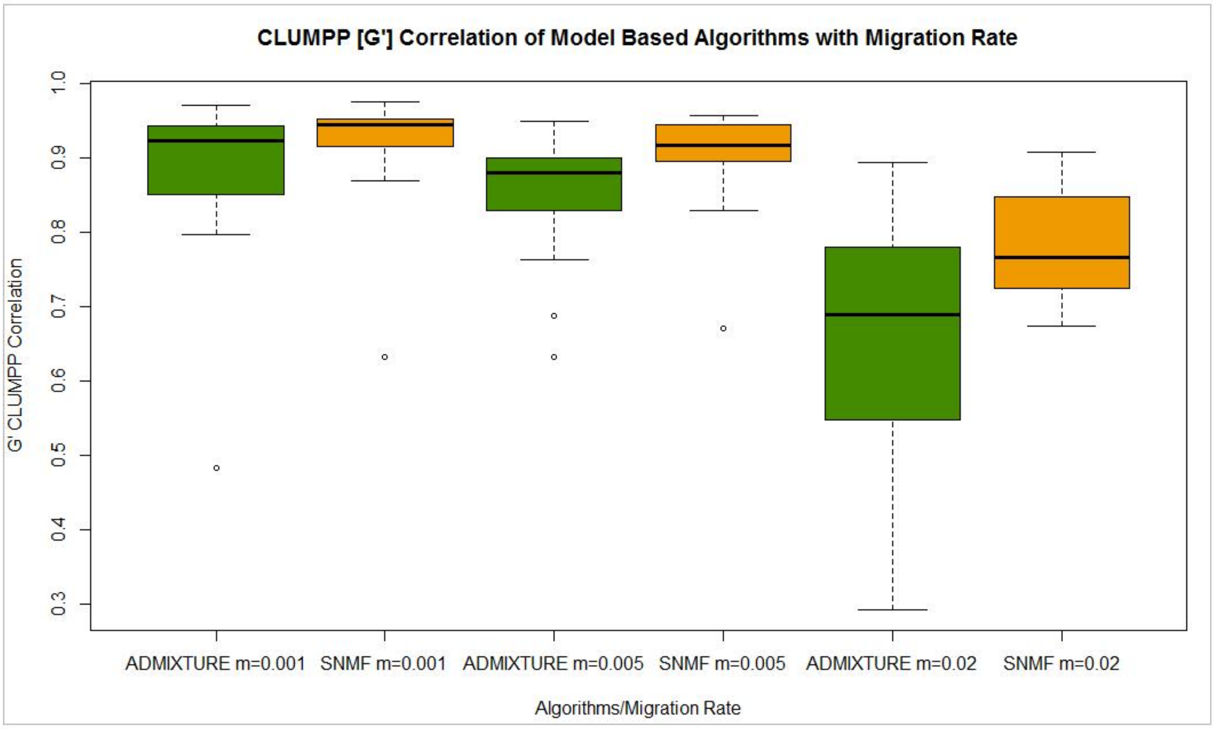


Figure 6.20: ADMIXTURE (green) and SNMF (orange) similarity coefficients (G') estimated by CLUMPP displayed by the three levels of migration rate and corroborating the best performance of SNMF in all cases and also ratifying the decreasing trend of both algorithms as migration rate grows.

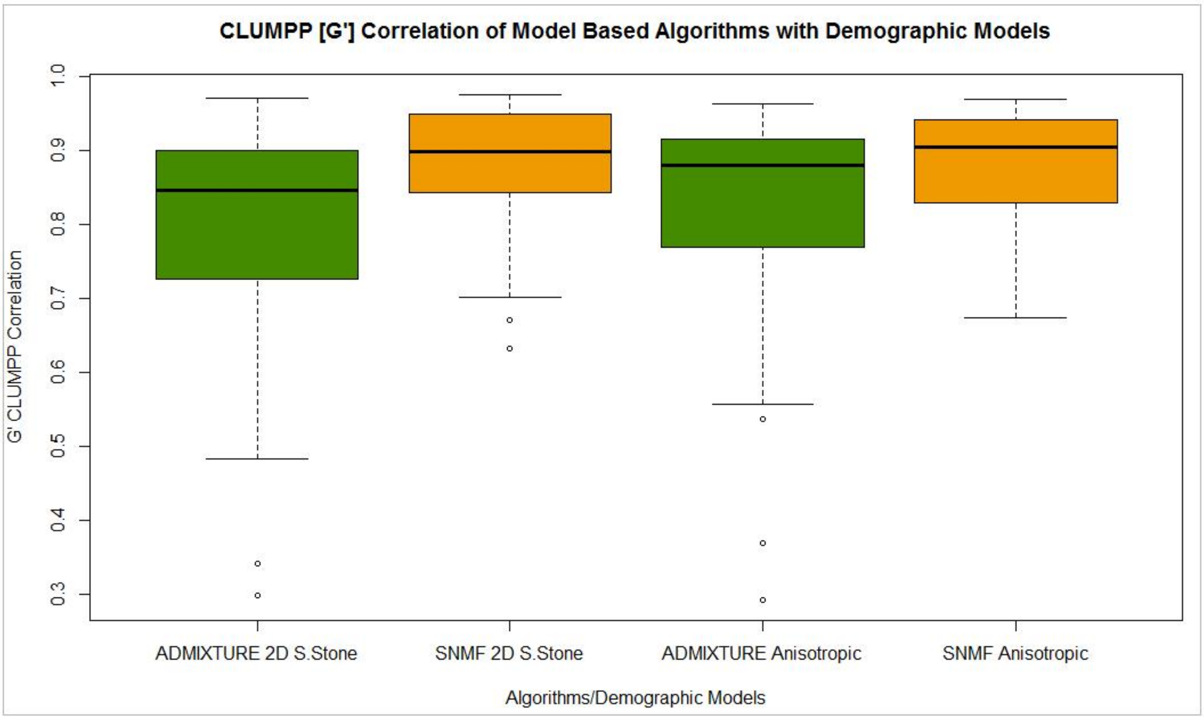


Figure 6.21: ADMIXTURE (green) and SNMF (orange) similarity coefficients (G') estimated by CLUMPP displayed by the two demographic models, corroborating the best performance of SNMF but not establishing a significant difference between demographic models.

Finally, aligning the CLUMPP similarity values (G') for all the 78 experimental datasets in a graph:

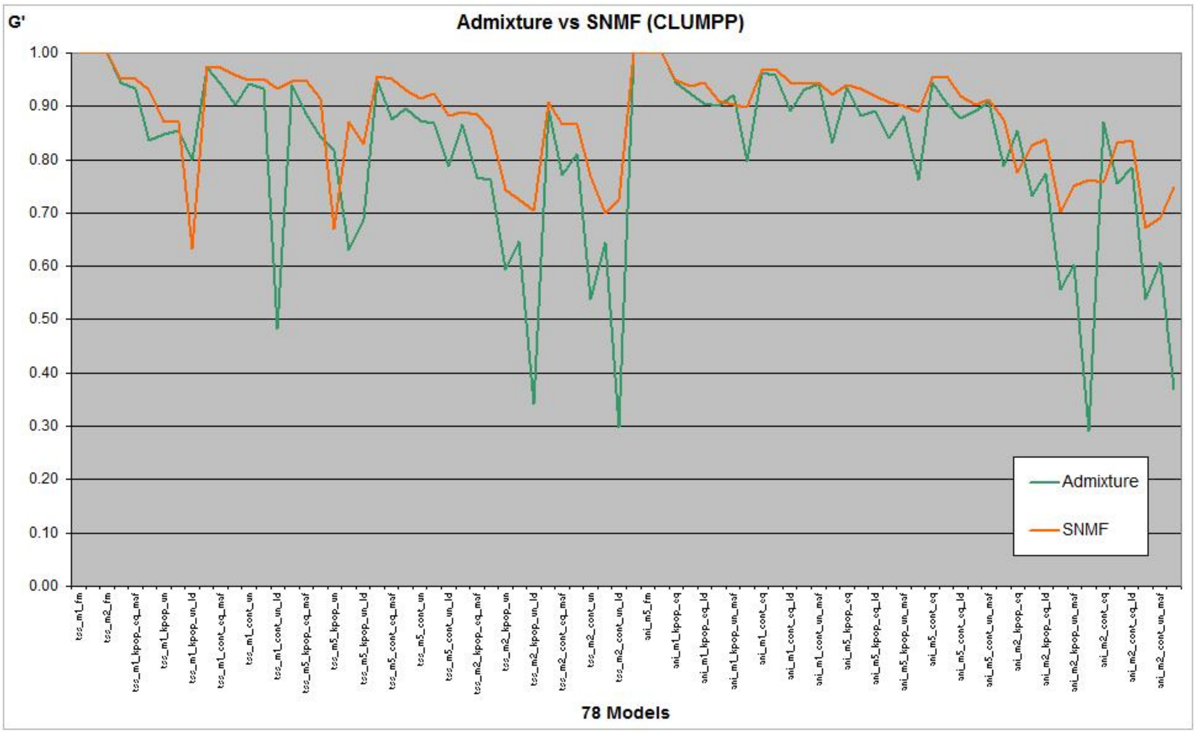


Figure 6.23: ADMIXTURE (green) and SNMF (orange) similarity coefficients (G') estimated by CLUMPP displayed along the 78 experimental models, corroborating again the best performance of SNMF algorithm.

7. Conclusions

The performance of five algorithms widely used in the field of population genetics for quantifying global population substructure has been tested under realistic spatial models. The validity and accuracy of the experimental model we have constructed is based on two pillars: first, the statistical predictive power of the 78 demographic scenarios designed and second, the robustness of Fastsimcoal2 performing coalescent simulations of DNA sequences. On the one hand, the set of demographic scenarios have been designed covering a broad range of sampling schemes. In the other hand, Fastsimcoal2 has been previously validated *[Yang 2014]* as highly scalable and flexible in simulating many different demographic histories and diverse DNA sequence structures such as SNPs.

Assuming that these initial conditions are well established, the analysis of the performance of commonly applied algorithms has determined that:

1) The five programs (PCA, MDS, SPA, ADMIXTURE and SNMF) show a strong robustness for detecting global ancestry in complex controlled geographic demographic scenarios. This is evidenced by the high degree of correlation between the estimated coordinates and the real geographical origin of individuals.
2) With regard to "Algorithmic" based methods, the best performing algorithm is smart-PCA (Eigensoft) since it shows the strongest level of correlation with the geographic sampling location and performing in a very fast and efficient way. Smart-PCA performance is followed very closely by MDS (PLINK) in terms of high correlation with the real geographical origin of individuals. A large number of computational tasks for processing SNP data can be easily performed via PLINK, and this is an additional advantage for performing global ancestry detection with the MDS-PLINK algorithm without changing the program platform. By using SPA, we have obtained a poorest correlation degree and we have experienced a shocking computational response time: while smart-PCA and MDS-PLINK solved the ancestry estimation of bigger datasets in minutes (2250 individuals - 944,000 SNPs), SPA required days with the same datasets.
3) With respect to "Model" based methods, the best performing algorithm is SNMF since it shows the highest degree of similarity between the different sampling cases and the reference datasets. In contrast, ADMIXTURE resulted in a lowest level of correlation between sampling and base datasets and, similarly to SPA, suffers from a disproportionate and severe response time that makes difficult to coordinate the different stages of the pipeline.
4) The migration rate level has a very significant impact on the validity of the results from the five algorithms: the higher the migration rate, the lower accuracy in the results. For instance, the best performing "algorithmic" program, smartPCA, experiences a Mantel correlation decreases from 0.97 when migration rate is m=0.001, to 0.86 when migration rate is 20 times greater

(m=0.02), in line with the principle which asserts that migration has a homogenizing effect on the genetic variation in populations.

5) The 2D stepping stone demographic model exhibits a slightly higher degree of correlation than anisotropic model for all algorithms. In average, the 2D stepping stone obtain Mantel and Procrustes correlations that are 6% and 4% above anisotropic model respectively.

6) The contagious sampling method performs slightly better than selecting populations randomly. For instance, for MDS-PLINK program under Procrustes test, the mean for the 36 contagious datasets is 0.90 while the mean for 36 two stepping stone models is 0.85 (6.4% below contagious). Similarly, for SNMF program under CLUMPP test, the mean for the 36 contagious datasets is 0.89 while the mean for the 36 two stepping stone models is 0.83 (5.8% below contagious).

7) The Equal sampling shows a stronger degree of correlation than unequal sampling method for all algorithms but this has to be taken with caution as the average number of individuals for unequal datasets are significantly lower than equal cases and this can partially explain the deviation.

8) The LD filtering method performs slightly better correlated than No Filtering strategy while MAF cleaning is the worst method. This conclusion is applicable to the whole experimental dataset pool.

9) The comparison between Mantel and Procrustes test is a question that can not go unnoticed: while Mantel test shows a total mean of 0.77 for the three algorithmic methods and for the 78 datasets (a total of 234 cases), Procrustes test exhibits a total mean of 0.88, 13% higher than Mantel. This can lead us to a very different conclusion: or a less demanding behavior for Procrustes test or a dysfunction in Mantel test amplifying artificially the decorrelation.

8. Appendices: Linux Shell Scripts and “R” Code

STEP 1 - SHELL SCRIPT FOR AUTOMATING FASTSIMCOAL2 AND ALGORITHMS ON BASE CASES

```
====================================================================================
pop_train1

##########################################################################
############## POPULATION ALGORITHM  #####################################
#Notes:
#FASTSIMCOAL2 installed at /home/ubuntu/fast and executable renamed as fsc
#ARLEQUIN2PLINK CONVERSOR installed at /home/ubuntu/dist
#ADMIXTURE installed at /home/ubuntu/admix
#SNMF installed at /home/ubuntu/snmf
#EIGENSOFT installed at /home/ubuntu/eigen
#SPA installed at /home/ubuntu/spa
#PLINK(MDS) installed at /home/ubuntu/plink
#TREEMIX installed at /home/ubuntu/tree
#R installed
#R package OriGen installed
############## Just to force linux format ################################
dos2unix $1.par
############## Running FASTSIMCOAL2 ######################################
cp $1.par /home/ubuntu/fast
cd /home/ubuntu/fast
## one simulation -n1, four cores and four baches
./fsc -i $1.par -n1 -c4 -B4
cd $1
cp $1_1_1.arp /home/ubuntu/dist
cd ..
mv $1.par /home/ubuntu/dist
rm -r $1
read -p "Process finished, ENTER to show files ........"
############## Converting Arlequin to Plink ##############################
cd /home/ubuntu/dist
mv $1_1_1.arp $1.arp
java -jar ConvertArlequinToPlink.jar $1.arp $1
############## Creating ped and Removing missing snp #####################
grep "[G|T|A|C] 0" $1.bim|awk '{ print $2}' > remove.snp
/home/ubuntu/plink/plink.107 --bfile $1 --recode --tab --out $1 --noweb
read -p "NOW EDIT WITH VI AND CHANGE SAMPLE 0  ........"
## tail -n +2 $1.ped > pedped2
## head -n 1 $1.ped > pedped1
## vi pedped1 CHANGE SAMPLE 0
## rm $1.ped
## cat pedped1 pedped2 > $1.ped
## rm pedped1 pedped2
## SPA is reluctant to run with missing values
/home/ubuntu/plink/plink.109 --file $1 --exclude remove.snp --noweb --make-bed --out $1bis
rm $1.bed $1.bim $1.fam
mv $1bis.bed $1.bed
mv $1bis.bim $1.bim
mv $1bis.fam $1.fam
############## Running ADMIXTURE #########################################
/home/ubuntu/admix/admixture $1.bed 4 -j4
## K ancestries = 4
mv $1.4.Q $1_admix.4.Q
rm $1.4.P
############## Running SNMF ##############################################
/home/ubuntu/snmf/bin/ped2geno $1.ped $1.geno
## K ancestries = 4
/home/ubuntu/snmf/bin/sNMF -x $1.geno -K 4
mv $1.4.Q $1_snmf.4.Q
rm $1.4.G
############## Running EIGENSOFT SMartPCA ################################
## just two principal components -k 2
read -p "CONTROL BEFORE PCA ........"
smartpca.perl -i $1.bed -a $1.bim -b $1.fam -o $1.pca -q NO -k 2 -p $1.pca -e $1.pca -l $1.pca
read -p "CONTROL AFTER PCA ........"
############## Running SPA ###############################################
#/home/ubuntu/spa/spa --bfile $1 --location-output $1.spa -r 0.0001
############## Running MDS ###############################################
## Pairwise IBS estimation
```

```
/home/ubuntu/plink/plink.109 --bfile $1 --genome --noweb --out $1
rm $1.log $1.nosex
/home/ubuntu/plink/plink.109 --bfile $1 --read-genome $1.genome --cluster --mds-plot 2 --noweb --out $1
############## Saving RESULTS ############################################
mkdir /home/ubuntu/results/$1
mv  $1.par $1.arp $1.bed $1.bim $1.fam $1_admix.4.Q $1_snmf.4.Q $1.ped $1.map $1.geno $1.pca.evec $1.spa $1.genome $1.mds $1.cluster*
/home/ubuntu/results/$1
rm $1.pca.* *.log *.nosex $1.pca remove.snp $1bis.* *.mod
cd /home/ubuntu/results/$1
read -p "Process finished, ENTER to show files ........"
ls -l
#######################################################################
```

STEP 2 - SHELL SCRIPT FOR AUTOMATING SAMPLING ON BASE CASES (POP,IND, MAF, LD)

```
================================================================================
pop_train2

mkdir /home/ubuntu/results/tss/tss_m2_cont_eq
mkdir /home/ubuntu/results/tss/tss_m2_cont_eq_ld
mkdir /home/ubuntu/results/tss/tss_m2_cont_eq_maf
mkdir /home/ubuntu/results/tss/tss_m2_cont_un
mkdir /home/ubuntu/results/tss/tss_m2_cont_un_ld
mkdir /home/ubuntu/results/tss/tss_m2_cont_un_maf
mkdir /home/ubuntu/results/tss/tss_m2_kpop_eq
mkdir /home/ubuntu/results/tss/tss_m2_kpop_eq_ld
mkdir /home/ubuntu/results/tss/tss_m2_kpop_eq_maf
mkdir /home/ubuntu/results/tss/tss_m2_kpop_un
mkdir /home/ubuntu/results/tss/tss_m2_kpop_un_ld
mkdir /home/ubuntu/results/tss/tss_m2_kpop_un_maf
mkdir /home/ubuntu/results/ani/ani_m2_cont_eq
mkdir /home/ubuntu/results/ani/ani_m2_cont_eq_ld
mkdir /home/ubuntu/results/ani/ani_m2_cont_eq_maf
mkdir /home/ubuntu/results/ani/ani_m2_cont_un
mkdir /home/ubuntu/results/ani/ani_m2_cont_un_ld
mkdir /home/ubuntu/results/ani/ani_m2_cont_un_maf
mkdir /home/ubuntu/results/ani/ani_m2_kpop_eq
mkdir /home/ubuntu/results/ani/ani_m2_kpop_eq_ld
mkdir /home/ubuntu/results/ani/ani_m2_kpop_eq_maf
mkdir /home/ubuntu/results/ani/ani_m2_kpop_un
mkdir /home/ubuntu/results/ani/ani_m2_kpop_un_ld
mkdir /home/ubuntu/results/ani/ani_m2_kpop_un_maf

read -p "ENTER to continue ........"

/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/tss/tss_m2_fm/tss_m2_fm --keep tss_kpop_eq.fam --make-bed --out
/home/ubuntu/results/tss/tss_m2_kpop_eq/tss_m2_kpop_eq --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/tss/tss_m2_fm/tss_m2_fm --keep tss_kpop_un.fam --make-bed --out
/home/ubuntu/results/tss/tss_m2_kpop_un/tss_m2_kpop_un --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/tss/tss_m2_fm/tss_m2_fm --keep tss_cont_eq.fam --make-bed --out
/home/ubuntu/results/tss/tss_m2_cont_eq/tss_m2_cont_eq --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/tss/tss_m2_fm/tss_m2_fm --keep tss_cont_un.fam --make-bed --out
/home/ubuntu/results/tss/tss_m2_cont_un/tss_m2_cont_un --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/ani/ani_m2_fm/ani_m2_fm --keep ani_kpop_eq.fam --make-bed --out
/home/ubuntu/results/ani/ani_m2_kpop_eq/ani_m2_kpop_eq --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/ani/ani_m2_fm/ani_m2_fm --keep ani_kpop_un.fam --make-bed --out
/home/ubuntu/results/ani/ani_m2_kpop_un/ani_m2_kpop_un --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/ani/ani_m2_fm/ani_m2_fm --keep ani_cont_eq.fam --make-bed --out
/home/ubuntu/results/ani/ani_m2_cont_eq/ani_m2_cont_eq --noweb
/home/ubuntu/plink/plink.109 --file /home/ubuntu/results/ani/ani_m2_fm/ani_m2_fm --keep ani_cont_un.fam --make-bed --out
/home/ubuntu/results/ani/ani_m2_cont_un/ani_m2_cont_un --noweb

read -p "Principal files created continue ........"
####/home/ubuntu/plink/plink.109 --bfile ani_kpop_eq --maf 0.00001 --make-bed --out ani_kpop_eq_000 --noweb
################ MAF generation ####################################
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_kpop_eq/tss_m2_kpop_eq --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/tss/tss_m2_kpop_eq_maf/tss_m2_kpop_eq_maf
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_kpop_un/tss_m2_kpop_un --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/tss/tss_m2_kpop_un_maf/tss_m2_kpop_un_maf
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_cont_eq/tss_m2_cont_eq --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/tss/tss_m2_cont_eq_maf/tss_m2_cont_eq_maf
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_cont_un/tss_m2_cont_un --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/tss/tss_m2_cont_un_maf/tss_m2_cont_un_maf
```

```
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_kpop_eq/ani_m2_kpop_eq --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/ani/ani_m2_kpop_eq_maf/ani_m2_kpop_eq_maf
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_kpop_un/ani_m2_kpop_un --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/ani/ani_m2_kpop_un_maf/ani_m2_kpop_un_maf
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_cont_eq/ani_m2_cont_eq --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/ani/ani_m2_cont_eq_maf/ani_m2_cont_eq_maf
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_cont_un/ani_m2_cont_un --noweb --maf 0.05 --make-bed --out
/home/ubuntu/results/ani/ani_m2_cont_un_maf/ani_m2_cont_un_maf

read -p "MAF done continue ........"
################## LD generation ########################################
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_kpop_eq/tss_m2_kpop_eq --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_kpop_eq/tss_m2_kpop_eq --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/tss/tss_m2_kpop_eq_ld/tss_m2_kpop_eq_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_kpop_un/tss_m2_kpop_un --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_kpop_un/tss_m2_kpop_un --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/tss/tss_m2_kpop_un_ld/tss_m2_kpop_un_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_cont_eq/tss_m2_cont_eq --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_cont_eq/tss_m2_cont_eq --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/tss/tss_m2_cont_eq_ld/tss_m2_cont_eq_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_cont_un/tss_m2_cont_un --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/tss/tss_m2_cont_un/tss_m2_cont_un --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/tss/tss_m2_cont_un_ld/tss_m2_cont_un_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_kpop_eq/ani_m2_kpop_eq --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_kpop_eq/ani_m2_kpop_eq --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/ani/ani_m2_kpop_eq_ld/ani_m2_kpop_eq_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_kpop_un/ani_m2_kpop_un --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_kpop_un/ani_m2_kpop_un --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/ani/ani_m2_kpop_un_ld/ani_m2_kpop_un_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_cont_eq/ani_m2_cont_eq --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_cont_eq/ani_m2_cont_eq --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/ani/ani_m2_cont_eq_ld/ani_m2_cont_eq_ld
rm /home/ubuntu/sampling/plink.*
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_cont_un/ani_m2_cont_un --noweb --indep 50 5 2
/home/ubuntu/plink/plink.109 --bfile /home/ubuntu/results/ani/ani_m2_cont_un/ani_m2_cont_un --noweb --extract plink.prune.in --make-bed --out
/home/ubuntu/results/ani/ani_m2_cont_un_ld/ani_m2_cont_un_ld
rm /home/ubuntu/sampling/plink.*

read -p "Process finished, ENTER to show files ........"
```

STEP 3 AND 4 - SHELL SCRIPT AUTOMATING ALL ALGORITHMS ON ALL SUBCASES

```
================================================================================
pop_train3

cd /home/ubuntu/results/ani/ani_m2_cont_eq
/home/ubuntu/pop_train4 ani_m2_cont_eq
cd /home/ubuntu/results/ani/ani_m2_cont_eq_ld
/home/ubuntu/pop_train4 ani_m2_cont_eq_ld
cd /home/ubuntu/results/ani/ani_m2_cont_eq_maf
/home/ubuntu/pop_train4 ani_m2_cont_eq_maf
cd /home/ubuntu/results/ani/ani_m2_cont_un
....................................
/home/ubuntu/pop_train4 tss_m2_kpop_un

================================================================================
pop_train4

############### Creating ped and Removing missing snp ######################
grep "[G|T|A|C] 0" $1.bim|awk '{ print $2}' > remove.snp
/home/ubuntu/plink/plink.109 --bfile $1 --recode --tab --out $1 --noweb
/home/ubuntu/plink/plink.109 --file $1 --exclude remove.snp --noweb --make-bed --out $1bis
rm $1.bed $1.bim $1.fam
mv $1bis.bed $1.bed
mv $1bis.bim $1.bim
mv $1bis.fam $1.fam
############### Running ADMIXTURE ###########################################
/home/ubuntu/admix/admixture $1.bed 4 -j4
mv $1.4.Q $1_admix.4.Q
rm $1.4.P
```

```
############## Running SNMF #############################################
/home/ubuntu/snmf/bin/ped2geno $1.ped $1.geno
/home/ubuntu/snmf/bin/sNMF -x $1.geno -K 4
mv  $1.4.Q $1_snmf.4.Q
rm $1.4.G
############## Running EIGENSOFT SMartPCA ###############################
smartpca.perl -i $1.bed -a $1.bim -b $1.fam -o $1.pca -q NO -k 2 -p $1.pca -e $1.pca -l $1.pca
############## Running SPA ##############################################
/home/ubuntu/spa/spa --bfile $1 --location-output $1.spa -r 0.0001
############## Running MDS ##############################################
/home/ubuntu/plink/plink.109 --bfile $1 --genome --noweb --out $1
rm $1.log $1.nosex
/home/ubuntu/plink/plink.109 --bfile $1 --read-genome $1.genome --cluster --mds-plot 2 --noweb --out $1
############## Saving RESULTS ###########################################
rm *.cluster? *.pca *.pca.par *.ps *.xtxt *.nosex *.log remove.snp $1bis.*
########################################################################
```

STEP 5 - R SCRIPT FOR GENERATING PLINK FAM FILES FOR FURTHER SAMPLING SUBCASES

```
================================================================================
ani_cont_eq.R

linies <- vector()
j<-1
kpop <-
sort(c(11,13,23,2,4,15,20,21,24,27,28,41,43,45,59,88,89,90,32,46,48,49,50,62,63,65,93,37,51,52,53,54,55,66,69,100,101,104,107,108,111,114,118,119,
120))
for (k in c(1:45))
  {
  for (i in (1:10))
                {
                linies[j] <- paste("Sample",kpop[k]," ",kpop[k]*10-11+i,sep="")
                j<-j+1}
                }
write(linies, file="/home/ubuntu/sampling/ani_cont_eq.fam")

================================================================================
ani_cont_un.R

linies <- vector()
kpop <-
sort(c(11,13,23,2,4,15,20,21,24,27,28,41,43,45,59,88,89,90,32,46,48,49,50,62,63,65,93,37,51,52,53,54,55,66,69,100,101,104,107,108,111,114,118,119,
120))
llevo <- 1
for (k in c(1:45))
  {j <- sample(1:5,1)
  ind <- sort(sample(c((kpop[k]*10-10):(kpop[k]*10-1)),j))
  for (i in (1:j))
                {linies[llevo] <- paste("Sample",kpop[k]," ",ind[i],sep="")
                llevo <- llevo+1}
  }
write(linies, file="/home/ubuntu/sampling/ani_cont_un.fam")


================================================================================
ani_kpop_eq.R

linies <- vector()
j <- 1
kpop <- sort(sample(c(1:125),45))
for (k in c(1:45))
  {
  for (i in (1:10))
                {
                linies[j] <- paste("Sample",kpop[k]," ",kpop[k]*10-11+i,sep="")
                if (linies[j] == "Sample1 0"){linies[j] <- "Sample1 A"}
                j<-j+1}
                }
write(linies, file="/home/ubuntu/sampling/ani_kpop_eq.fam")


================================================================================
ani_kpop_un.R

linies <- vector()
```

```
kpop <- sort(sample(c(1:125),45))
llevo <- 1
for (k in c(1:45))
  {j <- sample(1:5,1)
  ind <- sort(sample(c((kpop[k]*10-10):(kpop[k]*10-1)),j))
  for (i in (1:j))
                {linies[llevo] <- paste("Sample",kpop[k]," ",ind[i],sep="")
                if (linies[llevo] == "Sample1 0"){linies[llevo] <- "Sample1 A"}
                llevo <- llevo+1}
  }
write(linies, file="/home/ubuntu/sampling/ani_kpop_un.fam")

==============================================================================
tss_cont_eq.R

linies <- vector()
j <- 1
kpop <-
sort(sample(c(1,17,33,6,20,19,37,22,7,11,12,27,13,29,45,46,47,48,79,80,65,67,53,54,55,71,72,74,60,90,106,93,123,107,121,123,113,97,99,130,116,102
,10,118,119,138,153,168,154,155,156,172,157,174,175,147,162,165,164,179,196,212,183,184,200,214,217,187,219,190,191,206,210,223,224),75))
for (k in c(1:75))
  {
  for (i in (1:10))
                {
                linies[j] <- paste("Sample",kpop[k]," ",kpop[k]*10-11+i,sep="")
                if (linies[j] == "Sample1 0"){linies[j] <- "Sample1 A"}
                j<-j+1}
                }
write(linies, file="/home/ubuntu/sampling/tss_cont_eq.fam")

==============================================================================
tss_cont_un.R

linies <- vector()
kpop <-
sort(sample(c(1,17,33,6,20,19,37,22,7,11,12,27,13,29,45,46,47,48,79,80,65,67,53,54,55,71,72,74,60,90,106,93,123,107,121,123,113,97,99,130,116,102
,10,118,119,138,153,168,154,155,156,172,157,174,175,147,162,165,164,179,196,212,183,184,200,214,217,187,219,190,191,206,210,223,224),75))
llevo <- 1
for (k in c(1:75))
  {j <- sample(1:5,1)
  ind <- sort(sample(c((kpop[k]*10-10):(kpop[k]*10-1)),j))
  for (i in (1:j))
                {linies[llevo] <- paste("Sample",kpop[k]," ",ind[i],sep="")
                if (linies[llevo] == "Sample1 0"){linies[llevo] <- "Sample1 A"}
                llevo <- llevo+1}
  }
write(linies, file="/home/ubuntu/sampling/tss_cont_un.fam")


==============================================================================
tss_kpop_eq.R

linies <- vector()
j <- 1
kpop <- sort(sample(c(1:225),75))
for (k in c(1:75))
  {
  for (i in (1:10))
                {
                linies[j] <- paste("Sample",kpop[k]," ",kpop[k]*10-11+i,sep="")
                if (linies[j] == "Sample1 0"){linies[j] <- "Sample1 A"}
                j<-j+1}
                }
write(linies, file="/home/ubuntu/sampling/tss_kpop_eq.fam")
==============================================================================
tss_kpop_un.R

linies <- vector()
kpop <- sort(sample(c(1:225),75))
llevo <- 1
for (k in c(1:75))
  {j <- sample(1:5,1)
  ind <- sort(sample(c((kpop[k]*10-10):(kpop[k]*10-1)),j))
  for (i in (1:j))
                {linies[llevo] <- paste("Sample",kpop[k]," ",ind[i],sep="")
                if (linies[llevo] == "Sample1 0"){linies[llevo] <- "Sample1 A"}
                llevo <- llevo+1}
  }
write(linies, file="/home/ubuntu/sampling/tss_kpop_un.fam")
```

STEP 6 - R SCRIPT FOR APPLYING MANTEL AND PROCRUSTES ON PCA, MDS AND SPA

```
==============================================================================
mantel.R

library(vegan)
library(ade4)
library(prabclus)
conn <- file("/home/ubuntu/results/task",open="r")
linn <-readLines(conn)
theheader1 <-
"case,m_mds_cor,m_mds_sig,p_mds_cor,p_mds_sig,m_spa_cor,m_spa_sig,p_spa_cor,p_spa_sig,m_pca_cor,m_pca_sig,p_pca_cor,p_pca_sig,K4,K5,
K6,Gadm,Gsnm,Gas,NumInd,SNPs,BedSize"
write(theheader1,file="/home/ubuntu/results/res")
head1 <- "case                  fam    bed         mds     pca              spa         adm     snm              OK"
head2 <- "=================     ===          ===          ===     ===                ===          ===     ===                 ==="
write(head1,file="/home/ubuntu/results/rec")
write(head2,file="/home/ubuntu/results/rec",append="TRUE")

for (i in 1:length(linn))
{
print(linn[i])
theroot <- substr(linn[i],1,3)
thepath <- paste("/home/ubuntu/results/",theroot,"/",linn[i],"/",sep="")
if (substr(linn[i],1,3)=="ani"){base_global <- read.table("/home/ubuntu/results/base_ani.txt")} else{base_global <-
read.table("/home/ubuntu/results/base_tss.txt")}
colnames(base_global) <- c("pop","ind","x","y")
famfile <- list.files(path=thepath,pattern="*[a-z].fam")
if (length(famfile) == 0){a1<-0}else{a1<-1}
bedfile <- list.files(path=thepath,pattern="*.bed")
if (length(bedfile) == 0){a2<-0}else{a2<-1}
mdsfile <- list.files(path=thepath,pattern="*.mds")
if (length(mdsfile) == 0){a3<-0}else{a3<-1}
pcafile <- list.files(path=thepath,pattern="*.evec")
if (length(pcafile) == 0){a4<-0}else{a4<-1}
spafile <- list.files(path=thepath,pattern="*.spa")
if (length(spafile) == 0){a5<-0}else{a5<-1}
admfile <- list.files(path=thepath,pattern="*admix*")
if (length(admfile) == 0){a6<-0}else{a6<-1}
snmfile <- list.files(path=thepath,pattern="*snmf*")
if (length(snmfile) == 0){a7<-0}else{a7<-1}
aa <- a1+a2+a3+a4+a5+a6+a7
if (aa == 7)
{
case <- linn[i]
fam <- read.table(paste(thepath,famfile,sep=""))
fam <- fam[,1:2]
colnames(fam) <- c("pop","ind")
fam$ind[fam$ind == 0] <- "A"; fam$ind[fam$ind == "0"] <- "A"
base <- merge(base_global, fam, by=c("pop","ind"))
mds <- read.table(paste(thepath,case,".mds",sep=""),header=TRUE)
mds <- mds[,c(1,2,4,5)]
colnames(mds) <- c("pop","ind","mds_x","mds_y")
mds$ind[mds$ind == 0] <- "A"; mds$ind[mds$ind == "0"] <- "A"
base <- merge(base, mds, by=c("pop","ind"))
spa <- read.table(paste(thepath,case,".spa",sep=""))
spa <- spa[,c(1,2,7,8)]
colnames(spa) <- c("pop","ind","spa_x","spa_y")
spa$ind[spa$ind == 0] <- "A"; spa$ind[spa$ind == "0"] <- "A"
base <- merge(base, spa, by=c("pop","ind"))
pca <- read.table(paste(thepath,case,".pca.evec",sep=""))
pca <- pca[,c(1,2,3)]
pca$pop <- substr(pca[,1],1,as.numeric(regexpr(":", pca[,1]))-1)
pca$ind <- substr(pca[,1],as.numeric(regexpr(":", pca[,1]))+1,nchar(as.character(pca[,1])))
pca <- pca[,c(4,5,2,3)]
colnames(pca) <- c("pop","ind","pca_x","pca_y")
pca$ind[pca$ind == 0] <- "A"; pca$ind[pca$ind == "0"] <- "A"
base <- merge(base, pca, by=c("pop","ind"))
dbase <- coord2dist(coordmatrix=base[,3:4],file.format="decimal2")
dmds <- coord2dist(coordmatrix=base[,5:6],file.format="decimal2")
```

```
mant <- mantel(dbase,dmds)
prot <- protest(base[,3:4],base[,5:6])
m_mds_cor <- mant$statistic
m_mds_sig <- mant$signif
p_mds_cor <- prot$scale
p_mds_sig <- prot$signif
dspa <- coord2dist(coordmatrix=base[,7:8],file.format="decimal2")
mant <- mantel(dbase,dspa)
prot <- protest(base[,3:4],base[,7:8])
m_spa_cor <- mant$statistic
m_spa_sig <- mant$signif
p_spa_cor <- prot$scale
p_spa_sig <- prot$signif
dpca <- coord2dist(coordmatrix=base[,9:10],file.format="decimal2")
mant <- mantel(dbase,dpca)
prot <- protest(base[,3:4],base[,9:10])
m_pca_cor <- mant$statistic
m_pca_sig <- mant$signif
p_pca_cor <- prot$scale
p_pca_sig <- prot$signif
#############################################
### bestK results collection
#############################################
bestfile <- list.files(path="/home/ubuntu/bestK/",pattern=paste(case,".snm",sep=""))
if (length(bestfile) == 0)
{
K4 <- -1
K5 <- -1
K6 <- -1
}
else
{
bestK <-readLines(paste("/home/ubuntu/bestK/",bestfile,sep=""))
K4<-substr(bestK[1],regexpr("0.",bestK[1]),regexpr("0.",bestK[1])+6)
K5<-substr(bestK[2],regexpr("0.",bestK[2]),regexpr("0.",bestK[2])+6)
K6<-substr(bestK[3],regexpr("0.",bestK[3]),regexpr("0.",bestK[3])+6)
}
#############################################
### CLUMPP ADMIXTURE results collection
#############################################
clumfile <- list.files(path="/home/ubuntu/results/clumpp",pattern=paste(case,".adm",sep=""))
if (length(clumfile) == 0)
{Ga <- -1}
else
{
theG <-readLines(paste("/home/ubuntu/results/clumpp/",clumfile,sep=""))
Ga <- substr(theG[length(theG)],1,6)
}
#############################################
### CLUMPP SNMF results collection
#############################################
clumfile <- list.files(path="/home/ubuntu/results/clumpp",pattern=paste(case,".snm",sep=""))
if (length(clumfile) == 0)
{Gs <- -1}
else
{
theG <-readLines(paste("/home/ubuntu/results/clumpp/",clumfile,sep=""))
Gs <- substr(theG[length(theG)],1,6)
}
####################################################
### CLUMPP ADMIXTURE vs SNMF results collection
####################################################
clumfile <- list.files(path="/home/ubuntu/results/clumpp",pattern=paste(case,".as",sep=""))
if (length(clumfile) == 0)
{Gas <- -1}
else
{
theG <-readLines(paste("/home/ubuntu/results/clumpp/",clumfile,sep=""))
Gas <- substr(theG[length(theG)],1,6)
}
#############################################
## obtaining individuals, snps, bed file size
#############################################
famfile <- list.files(path=thepath,pattern="*[a-z].fam")
if (length(famfile) == 0)
```

```
{numind <- -1}
else
{
fam <- read.table(paste(thepath,famfile,sep=""))
numind <- nrow(fam)
}

bimfile <- list.files(path=thepath,pattern="*[a-z].bim")
if (length(bimfile) == 0)
{SNPS <- -1}
else
{
bim <- read.table(paste(thepath,bimfile,sep=""))
SNPS <- nrow(bim)
}

bedfile <- list.files(path=thepath,pattern="*[a-z].bed")
if (length(bedfile) == 0)
{bedsiz <- -1}
else
{
bedsiz <- file.size(paste(thepath,bedfile,sep=""))/1000000
}
##############################################

results1 <-
paste(case,m_mds_cor,m_mds_sig,p_mds_cor,p_mds_sig,m_spa_cor,m_spa_sig,p_spa_cor,p_spa_sig,m_pca_cor,m_pca_sig,p_pca_cor,p_pca_sig,K
4,K5,K6,Ga,Gs,Gas,numind,SNPS,bedsiz,sep=",")
write(t(results1),file="/home/ubuntu/results/res",append=TRUE)
}
else
{
case <- linn[i]
case <- paste(case,"             ",sep="")
case <- substr(case,1,20)
recue <- paste(case,a1,a2,a3,a4,a5,a6,a7,aa,sep="\t")
results1 <- paste(case,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,sep=",")
write(results1,file="/home/ubuntu/results/res",append=TRUE)
write(recue,file="/home/ubuntu/results/rec",append=TRUE)
}
}
close(conn)
```

STEP 7- SHELL SCRIPTS FOR ESTIMATING BEST K NUMBER OF ANCESTRIES

```
================================================================================
mixK
================================================================================
for K in 4 5 6
do
echo K=$K
/home/ubuntu/admix/admixture --cv /home/ubuntu/results/$2/$1/$1.bed $K -j4 > $1.$K.mix.log
done
grep -h CV $1*.mix.log > $1.mix
rm *.mix.log
cat $1.mix
================================================================================
snmfK

cd /home/ubuntu/bestK
cp /home/ubuntu/results/$2/$1/$1.geno .
for K in 4 5 6
do
echo K=$K; /home/ubuntu/snmf/bin/sNMF -x $1.geno -p 4 -K $K -c > $1.$K.log
done
rm *.G *.Q $1.geno $1_I.geno
grep "Cross-Entropy (masked data):" $1*.log > $1.snm
rm $1*.log
cat $1.snm
================================================================================
```

STEP 8 - R SCRIPT FOR APPLYING CLUMPP ON ADMIXTURE VS BASE CASES

```
================================================================================
clum_admix.R

library(vegan)
library(ade4)
library(prabclus)
conn <- file("/home/ubuntu/results/task2",open="r")
linn <-readLines(conn)

for (i in 1:length(linn))
{
print(linn[i])
theroot <- substr(linn[i],1,3)
thepath <- paste("/home/ubuntu/results/",theroot,"/",linn[i],"/",sep="")

famfile <- list.files(path=thepath,pattern="*[a-z].fam")
if (length(famfile) == 0){a1<-0}else{a1<-1}
admfile <- list.files(path=thepath,pattern="*admix*")
if (length(admfile) == 0){a6<-0}else{a6<-1}
aa <- a1+a6
if (aa == 2)
{
case <- linn[i]
fam <- read.table(paste(thepath,famfile,sep=""))
fam <- fam[,2]
adm <- read.table(paste(thepath,admfile,sep=""))
adm <- cbind(fam,adm)
print(adm[1,1])
adm[,1] <- as.character(adm[,1])
if (adm[1,1]=="A"){adm[1,1]<-"9999"}
if (adm[1,1]=="0"){adm[1,1]<-"9999"}
###### READ BASE SNMF FILE ##############################
baseadm <- paste(substr(case,1,7),"fm",sep="")
thepathh <- paste("/home/ubuntu/results/",theroot,"/",baseadm,"/",sep="")
basefamfile <- list.files(path=thepathh,pattern="*[a-z].fam")
basefam <- read.table(paste(thepathh,basefamfile,sep=""))
basefam <- basefam[,2]
basefile <- list.files(path=thepathh,pattern="*admix*")
baseadm <- read.table(paste(thepathh,basefile,sep=""))
baseadm <- cbind(basefam,baseadm)
baseadm$basefam <- as.character(baseadm$basefam)
if (baseadm[1,1]=="A"){baseadm[1,1]<-"9999"}
if (baseadm[1,1]=="0"){baseadm[1,1]<-"9999"}
baseadm2 <- baseadm[baseadm$basefam %in% adm$fam,]
adm <- cbind(adm[,1],adm[,1],adm[,1],adm[,1],adm)
baseadm2 <- cbind(baseadm2[,1],baseadm2[,1],baseadm2[,1],baseadm2[,1],baseadm2)
adm[,5]<- ":"
baseadm2[,5]<- ":"
adm[,3]<- paste("(",adm[,3],")",sep="")
baseadm2[,3]<- paste("(",baseadm2[,3],")",sep="")

rownames(adm) <- NULL
rownames(baseadm2) <- NULL
colnames(adm) <- NULL
colnames(baseadm2) <- NULL
#adm[,1]<- adm[,1]+1
#adm[,2]<- adm[,2]+1
#adm[,3]<- adm[,3]+1
#adm[,4]<- adm[,4]+1
#baseadm2[,1] <- as.numeric(baseadm2[,1])
#baseadm2[,2] <- as.numeric(baseadm2[,2])
#baseadm2[,3] <- as.numeric(baseadm2[,3])
#baseadm2[,4] <- as.numeric(baseadm2[,4])
#baseadm2[,1]<- baseadm2[,1]+1
#baseadm2[,2]<- baseadm2[,2]+1
#baseadm2[,3]<- baseadm2[,3]+1
#baseadm2[,4]<- baseadm2[,4]+1
write.table(adm,"unouno",sep="\t",row.names=FALSE,quote=FALSE)
write.table(baseadm2,"dosdos",sep="\t",row.names=FALSE,quote=FALSE)
system(paste("cat unouno dosdos > thepop",sep=""))
system("rm unouno dosdos")
prepcommand <- "/home/ubuntu/clum/CLUMPP /home/ubuntu/clum/paramfile -i thepop "
theoptions <- paste(" -c ",nrow(adm),sep="")
prepcommand <- paste(prepcommand,theoptions,sep="")
print (prepcommand)
system(prepcommand, intern=TRUE,wait=TRUE)
system(paste("mv loveo.miscfile ",case,".adm",sep=""))
##system("rm loveo.outfile thepop")
}
else
{
}
}
system("tail -v -n 1 *.adm > CLUMPP.admix")
system("rm *.adm")
close(conn)
```

STEP 9 - R SCRIPT FOR APPLYING WILCOXON AND BOXPLOTS TO FINAL RESULTS

```
=================================================================================
wilcox.R

res1 <- read.table("c:/vic/pop/wilcoxon_2DSS.csv",header=TRUE,sep=";")
boxplot(res1[,c(9,3,6,10,4,7,11,5,8)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3","grey","tomato","royalblue3"),
 names=c("PCA m=0.001","MDS m=0.001","SPA m=0.001","PCA m=0.005","MDS m=0.005","SPA m=0.005","PCA m=0.02","MDS m=0.02","SPA
m=0.02"),
 ylab ="Mantel Correlation", xlab ="Algorithms/Migration Rate",
 main ="2D Steeping Stone - Mantel Correlation with Geographic sampling site"
  )
boxplot(res1[,c(18,12,15,19,13,16,20,14,17)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA m=0.001","MDS m=0.001","SPA m=0.001","PCA m=0.005","MDS m=0.005","SPA m=0.005","PCA m=0.02","MDS
m=0.02","SPA m=0.02"),
                ylab ="Procrustes Correlation", xlab ="Algorithms/Migration Rate",
                main ="2D Steeping Stone - Procrustes Correlation with Geographic sampling site"
)

res2 <- read.table("c:/vic/pop/wilcoxon_ANIS.csv",header=TRUE,sep=";")
boxplot(res2[,c(9,3,6,10,4,7,11,5,8)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA m=0.001","MDS m=0.001","SPA m=0.001","PCA m=0.005","MDS m=0.005","SPA m=0.005","PCA m=0.02","MDS
m=0.02","SPA m=0.02"),
                ylab ="Mantel Correlation", xlab ="Algorithms/Migration Rate",
                main ="ANISOTROPIC - Mantel Correlation with Geographic sampling site"
)
boxplot(res2[,c(18,12,15,19,13,16,20,14,17)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA m=0.001","MDS m=0.001","SPA m=0.001","PCA m=0.005","MDS m=0.005","SPA m=0.005","PCA m=0.02","MDS
m=0.02","SPA m=0.02"),
                ylab ="Procrustes Correlation", xlab ="Algorithms/Migration Rate",
                main ="ANISOTROPIC - Procrustes Correlation with Geographic sampling site"
)

wilcox.test(res1$pca_man_001,res1$mds_man_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$pca_man_001,res1$spa_man_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$mds_man_001,res1$spa_man_001,paired=TRUE,alternative="greater")$p.value

wilcox.test(res1$pca_man_005,res1$mds_man_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$pca_man_005,res1$spa_man_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$mds_man_005,res1$spa_man_005,paired=TRUE,alternative="greater")$p.value

wilcox.test(res1$pca_man_02,res1$mds_man_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$pca_man_02,res1$spa_man_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$mds_man_02,res1$spa_man_02,paired=TRUE,alternative="greater")$p.value

wilcox.test(res1$pca_pro_001,res1$mds_pro_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$pca_pro_001,res1$spa_pro_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$mds_pro_001,res1$spa_pro_001,paired=TRUE,alternative="greater")$p.value

wilcox.test(res1$pca_pro_005,res1$mds_pro_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$pca_pro_005,res1$spa_pro_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$mds_pro_005,res1$spa_pro_005,paired=TRUE,alternative="greater")$p.value

wilcox.test(res1$pca_pro_02,res1$mds_pro_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$pca_pro_02,res1$spa_pro_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res1$mds_pro_02,res1$spa_pro_02,paired=TRUE,alternative="greater")$p.value



wilcox.test(res2$pca_man_001,res2$mds_man_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_man_001,res2$spa_man_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$mds_man_001,res2$spa_man_001,paired=TRUE,alternative="greater")$p.value

wilcox.test(res2$pca_man_005,res2$mds_man_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_man_005,res2$spa_man_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$mds_man_005,res2$spa_man_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_man_02,res2$mds_man_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_man_02,res2$spa_man_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$mds_man_02,res2$spa_man_02,paired=TRUE,alternative="greater")$p.value

wilcox.test(res2$pca_pro_001,res2$mds_pro_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_pro_001,res2$spa_pro_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$mds_pro_001,res2$spa_pro_001,paired=TRUE,alternative="greater")$p.value

wilcox.test(res2$pca_pro_005,res2$mds_pro_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_pro_005,res2$spa_pro_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$mds_pro_005,res2$spa_pro_005,paired=TRUE,alternative="greater")$p.value

wilcox.test(res2$pca_pro_02,res2$mds_pro_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$pca_pro_02,res2$spa_pro_02,paired=TRUE,alternative="greater")$p.value
wilcox.test(res2$mds_pro_02,res2$spa_pro_02,paired=TRUE,alternative="greater")$p.value



res3 <- read.table("c:/vic/pop/2DSS_ANIS.csv",header=TRUE,sep=";")
boxplot(res3[,c(9,1,5,10,2,6)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA 2DSS","MDS 2DSS","SPA 2DSS","PCA Aniso","MDS Aniso","SPA Aniso"),
                ylab ="Mantel Correlation", xlab ="Algorithms/Demographic Model",
                main ="Mantel Correlation with Demographic Model"
)
boxplot(res3[,c(11,3,7,12,4,8)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA 2DSS","MDS 2DSS","SPA 2DSS","PCA Aniso","MDS Aniso","SPA Aniso"),
                ylab ="Procrustes Correlation", xlab ="Algorithms/Demographic Model",
                main ="Procrustes Correlation with Demographic Model"
)

wilcox.test(res3$pca_man_tss,res3$pca_man_ani,paired=TRUE,alternative="greater")$p.value
wilcox.test(res3$mds_man_tss,res3$mds_man_ani,paired=TRUE,alternative="greater")$p.value
wilcox.test(res3$spa_man_tss,res3$spa_man_ani,paired=TRUE,alternative="greater")$p.value
wilcox.test(res3$pca_pro_tss,res3$pca_pro_ani,paired=TRUE,alternative="greater")$p.value
wilcox.test(res3$mds_pro_tss,res3$mds_pro_ani,paired=TRUE,alternative="greater")$p.value
wilcox.test(res3$spa_pro_tss,res3$spa_pro_ani,paired=TRUE,alternative="greater")$p.value

res4 <- read.table("c:/vic/pop/random_cont.csv",header=TRUE,sep=";")
boxplot(res4[,c(9,1,5,10,2,6)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
        names=c("PCA Contagious","MDS Contagious","SPA Contagious","PCA Random","MDS Random","SPA Random"),
                ylab ="Mantel Correlation", xlab ="Algorithms/Population Sampling Method",
                main ="Mantel Correlation with Population Sampling Method"
)
boxplot(res4[,c(11,3,7,12,4,8)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA Contagious","MDS Contagious","SPA Contagious","PCA Random","MDS Random","SPA Random"),
                ylab ="Procrustes Correlation", xlab ="Algorithms/Population Sampling Method",
                main ="Procrustes Correlation with Population Sampling Method"
)

wilcox.test(res4$pca_man_con,res4$pca_man_rnd,paired=TRUE,alternative="greater")$p.value
wilcox.test(res4$mds_man_con,res4$mds_man_rnd,paired=TRUE,alternative="greater")$p.value
wilcox.test(res4$spa_man_con,res4$spa_man_rnd,paired=TRUE,alternative="greater")$p.value
wilcox.test(res4$pca_pro_con,res4$pca_pro_rnd,paired=TRUE,alternative="greater")$p.value
wilcox.test(res4$mds_pro_con,res4$mds_pro_rnd,paired=TRUE,alternative="greater")$p.value
wilcox.test(res4$spa_pro_con,res4$spa_pro_rnd,paired=TRUE,alternative="greater")$p.value


res5 <- read.table("c:/vic/pop/equal_unequal.csv",header=TRUE,sep=";")
boxplot(res5[,c(9,1,5,10,2,6)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA Equal","MDS Equal","SPA Equal","PCA Unequal","MDS Unequal","SPA Unequal"),
                ylab ="Mantel Correlation", xlab ="Algorithms/Individuals Sampling Method",
                   main ="Mantel Correlation with Individuals Sampling Method"
)
boxplot(res5[,c(11,3,7,12,4,8)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
                names=c("PCA Equal","MDS Equal","SPA Equal","PCA Unequal","MDS Unequal","SPA Unequal"),
                ylab ="Procrustes Correlation", xlab ="Algorithms/Individuals Sampling Method",
                main ="Procrustes Correlation with Individuals Sampling Method"
)

wilcox.test(res5$pca_man_eq,res5$pca_man_un,paired=TRUE,alternative="greater")$p.value
wilcox.test(res5$mds_man_eq,res5$mds_man_un,paired=TRUE,alternative="greater")$p.value
wilcox.test(res5$spa_man_eq,res5$spa_man_un,paired=TRUE,alternative="greater")$p.value
wilcox.test(res5$pca_pro_eq,res5$pca_pro_un,paired=TRUE,alternative="greater")$p.value
wilcox.test(res5$mds_pro_eq,res5$mds_pro_un,paired=TRUE,alternative="greater")$p.value
wilcox.test(res5$spa_pro_eq,res5$spa_pro_un,paired=TRUE,alternative="greater")$p.value
```

```
res6 <- read.table("c:/vic/pop/MAF_LD_NO.csv",header=TRUE,sep=";")
boxplot(res6[,c(13,1,7,14,2,8,15,3,9)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3","grey","tomato","royalblue3"),
        names=c("PCA LD","MDS LD","SPA LD","PCA MAF","MDS MAF","SPA MAF","PCA NoFilt","MDS NoFilt","SPA NoFilt"),
        ylab ="Mantel Correlation", xlab ="Algorithms/Filtering Method",
        main ="Mantel Correlation with Filtering Method"
)

boxplot(res6[,c(16,4,10,17,5,11,18,6,12)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3","grey","tomato","royalblue3"),
        names=c("PCA LD","MDS LD","SPA LD","PCA MAF","MDS MAF","SPA MAF","PCA NoFilt","MDS NoFilt","SPA NoFilt"),
        ylab ="Procrustes Correlation", xlab ="Algorithms/Filtering Method",
        main ="Procrustes Correlation with Filtering Method"
)


boxplot(res6[,c(11,3,7,12,4,8)],col=c("grey","tomato","royalblue3","grey","tomato","royalblue3"),
        names=c("PCA Equal","MDS Equal","SPA Equal","PCA Unequal","MDS Unequal","SPA Unequal"),
        ylab ="Procrustes Correlation", xlab ="Algorithms/Individuals Sampling Method",
        main ="Procrustes Correlation with Individuals Sampling Method"
)

wilcox.test(res6$pca_man_ld,res6$pca_man_maf,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$pca_man_ld,res6$pca_man_no,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$mds_man_ld,res6$mds_man_maf,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$mds_man_ld,res6$mds_man_no,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$spa_man_ld,res6$spa_man_maf,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$spa_man_ld,res6$spa_man_no,paired=TRUE,alternative="greater")$p.value

wilcox.test(res6$pca_pro_ld,res6$pca_pro_maf,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$pca_pro_ld,res6$pca_pro_no,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$mds_pro_ld,res6$mds_pro_maf,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$mds_pro_ld,res6$mds_pro_no,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$spa_pro_ld,res6$spa_pro_maf,paired=TRUE,alternative="greater")$p.value
wilcox.test(res6$spa_pro_ld,res6$spa_pro_no,paired=TRUE,alternative="greater")$p.value

########################################################################################
##  CLUMPP on ADMIXTURE and SNMF
########################################################################################
res7 <- read.table("c:/vic/pop/clumpp1.csv",header=TRUE,sep=";")
boxplot(res7[,c(1,4,2,5,3,6)],col=c("chartreuse4","orange2","chartreuse4","orange2","chartreuse4","orange2"),
        names=c("ADMIXTURE m=0.001","SNMF m=0.001","ADMIXTURE m=0.005","SNMF m=0.005","ADMIXTURE m=0.02","SNMF m=0.02"),
        ylab ="G' CLUMPP Correlation", xlab ="Algorithms/Migration Rate", cex=0.7,
        main ="CLUMPP [G'] Correlation of Model Based Algorithms with Migration Rate"
)

res8 <- read.table("c:/vic/pop/clumpp2.csv",header=TRUE,sep=";")
boxplot(res8[,c(1,3,2,4)],col=c("chartreuse4","orange2","chartreuse4","orange2"),
        names=c("ADMIXTURE 2D S.Stone","SNMF 2D S.Stone","ADMIXTURE Anisotropic","SNMF Anisotropic"),
        ylab ="G' CLUMPP Correlation", xlab ="Algorithms/Demographic Models", cex=0.7,
        main ="CLUMPP [G'] Correlation of Model Based Algorithms with Demographic Models"
)

wilcox.test(res7$snmf_001,res7$adm_001,paired=TRUE,alternative="greater")$p.value
wilcox.test(res7$snmf_005,res7$adm_005,paired=TRUE,alternative="greater")$p.value
wilcox.test(res7$snmf_02,res7$adm_02,paired=TRUE,alternative="greater")$p.value

wilcox.test(res8$adm_ani,res8$adm_tss,paired=TRUE,alternative="greater")$p.value
wilcox.test(res8$snmf_ani,res8$snmf_tss,paired=TRUE,alternative="greater")$p.value
```

9. References

Alexander, D., & Lange, K. (2011). **Enhancements to the ADMIXTURE algorithm for individual ancestry estimation.** BMC Bioinformatics, 12, 246.

Alexander, D., Novembre, J., & Lange, K. (2009). **Fast model-based estimation of ancestry in unrelated individuals**. Genome Research, 19(9), 1655-64.

Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, & David Reich. (2006). **Principal components analysis corrects for stratification in genome-wide association studies**. Nature Genetics, 38(8), 904.

Cox, J., & Durrett, R. (2002). **The Stepping Stone Model: New Formulas Expose Old Myths**. The Annals of Applied Probability, 12(4), 1348-1377.

Diniz-Filho, J., Soares, T., Lima, J., Dobrovolski, R., Landeiro, V., De Campos Telles, M.,Bini, L. (2013). **Mantel test in population genetics**. Genetics and Molecular Biology, 36(4), 475-85.

Excoffier, L., & Foll, M. (2011). **Fastsimcoal: A continuous-time coalescent simulator of genomic diversity under arbitrarily complex evolutionary scenarios**. Bioinformatics, 27(9), 1332-1334.

Frichot, E., Mathieu, F., Trouillon, T., Bouchard, G., & François, O. (2014). **Fast and efficient estimation of individual ancestry coefficients**. Genetics, 196(4), 973-83.

Gao, X., & Martin, E. (2009). **Using Allele Sharing Distance for Detecting Human Population Stratification.** Human Heredity, 68(3), 182-191.

Hartl, D., & Clark, Andrew G. (2007). **Principles of population genetics** (4th ed.). Sunderland, Mass.: Sinauer Associates.

Jay, F., Sjödin, P., Jakobsson, M., & Blum, M. (2013). **Anisotropic Isolation by Distance: The Main Orientations of Human Genetic Differentiation**. Molecular Biology and Evolution, 30(3), 513-525.

Jobling, M. (2014). **Human evolutionary genetics** (2nd ed.). New York: Garland Science.

Kimura, M., & Weiss, G. (1964). **The Stepping Stone Model of Population Structure and the Decrease of Genetic Correlation with Distance**. Genetics,49(4), 561-76.

Kullback, S., & Leibler, R. (1951). **On Information and Sufficiency**. The Annals of Mathematical Statistics, 22(1), 79-86.

Lao, Altena, Becker, Brauer, Kraaijenbrink, Van Oven,..Kayser. (2013). **Clinal distribution of human genomic diversity across the Netherlands despite archaeological evidence for genetic discontinuities in Dutch population history**. Investigative Genetics, 4(1), 9.

Lao, Lu, Nothnagel, Junge, Freitag-Wolf, Caliebe,...Kayser. (2008). **Correlation between Genetic and Geographic Structure in Europe**. Current Biology, 18(16), 1241-1248.

Lavanya Rishishwar, Andrew B. Conley, Charles H. Wigington, Lu Wang, Augusto Valderrama-Aguirre, & I. King Jordan. (2015). **Ancestry, admixture and fitness in Colombian genomes**. Scientific Reports, 5, Scientific Reports, 2015, Vol.5.

Lee, Daniel D., & Seung, H. Sebastian. (1999). **Learning the parts of objects by non-negative matrix factorization**. Nature, 401(6755), 788.

Liu, Y., Nyunoya, T., Leng, S., Belinsky, S., Tesfaigzi, Y., & Bruse, S. (2013). **Softwares and methods for estimating genetic ancestry in human populations**. Human Genomics, 7, 1.

Ma, J., Amos, C., & You, M. (2012). **Principal Components Analysis of Population Admixture** (Principal Components Analysis of Admixture). PLoS ONE, 7(7), E40115.

Marjoram, P., & Wall, J. (2006). **Fast "coalescent" simulation.** BMC Genetics, 7, 16.

McVean, G., & Przeworski, M. (2009**). A Genealogical Interpretation of Principal Components Analysis** (Gene Genealogies and PCA). PLoS Genetics, 5(10), E1000686.

Novembre John, & Matthew Stephens. (2008). **Interpreting principal component analyses of spatial population genetic variation.** Nature Genetics,40(5), 646.

Padhukasahasram, B. (2014). **Inferring ancestry from population genomic data and its applications**. Frontiers in Genetics, 5, Frontiers in Genetics, 2014, Vol.5.

Patterson, N., Price, A., Reich, D., & Allison, D. (2006). **Population Structure and Eigenanalysis** (Population Structure and Eigenanalysis). PLoS Genetics, 2(12), E190.

Peña D. (2002) **Análisis de Datos Multivariantes,** McGraw Hill

Pritchard, J., Falush, D., Stephens, M. (2000). **Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies**. Genetics, 164(4), 1567-87.

Rañola, J., Novembre, J., & Lange, K. (2014). **Fast spatial ancestry via flexible allele frequency surfaces**. Bioinformatics (Oxford, England), 30(20), 2915-22.

Rosenberg, Noah A., Pritchard, Jonathan K., Weber, James L., Cann, Howard M., Kidd, Kenneth K., Zhivotovsky, Lev A., & Feldman, Marcus W. (2002). **Genetic structure of human populations**. (Reports). Science, 298(5602), 2381.

Rosenberg, N., Jakobsson, M, (2007). **CLUMPP: A cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure**. Bioinformatics, 23(14), 1801-1806.

Sham, Pc, Purcell, S, Neale, B, Toddbrown, K, Thomas, L, Ferreira, Mar, . . . Daly, Mj. (2007). **PLINK: A tool set for whole-genome association and population-based linkage analyses**. 81(3), 559-575.

Slatkin, M., & Maruyama, T. (1975). **Genetic drift in a cline**. Genetics, 81(1), 209-22.

Sokal, R., Oden, N., Thomson, B., & Novembre, J. (2012). **A Problem with Synthetic Maps**/Commentary. Human Biology, 84(5), 607-21.

Sterns, Stephen (2010). "**The Origin and Maintenance of Genetic Variation**, Principles of Evolution, Ecology and Behavior, Lecture 6 ", Yale University

Wang, Z.A., Szpiech, Z.A., Degnan, J.H., Jakobsson, Mattias, Pemberton, T.J., Hardy, J.A., . . . Rosenberg, N.A. (2010). **Comparing spatial maps of human population-genetic variation using Procrustes analysis**. Statistical Applications In Genetics And Molecular Biology, 9(1), E13.

Wen-Yun Yang, John Novembre, Eleazar Eskin, & Eran Halperin. (2012). **A model-based approach for analysis of spatial structure in genetic data**. Nature Genetics, 44(6), 725.

Wollstein, A., & Lao, O. (2015). **Detecting individual ancestry in the human genome**. Investigative Genetics, 6, 7.

Yang, T., Deng, H., & Niu, T. (2014). **Critical assessment of coalescent simulators in modeling recombination hotspots in genomic sequences**. BMC Bioinformatics, 15, 3.

Yang, W., Platt, A., Chiang, C., Eskin, E., Novembre, J., & Pasaniuc, B. (2014). **Spatial localization of recent ancestors for admixed individuals**. G3 (Bethesda, Md.), 4(12), 2505-18.