M2 Data Science : Statistics for Genomics and Health

Projet

A genetic atlas of human admixture History

S. Amoyal, O. Guedj, T. Marcoux Pépin

20/01/2020

Table des matières

Table des figures

1 Introduction

Les liens existant entre histoire des origines humaines, mouvement des populations et patrimoine génétique sont des sujets très étudiés notemment dans l'article *A genetic atlas of human admixture history*[1].

L'idée centrale de cette étude repose sur le fait qu'en s'étendant sur les différents continents, les populations se regroupent naturellement en sous-populations. Ces échanges et mouvements de populations entraînent une légère caractérisation des séquences d'ADN des individus appartenant a un groupe. Lorsque ces sous-populations se regroupent de nouveau, une "trace" est perceptible dans leur code génétique : c'est ce que les auteurs appellent *a genetic admixture*. Le but de l'étude était d'utiliser le materiel génétique d'une population multi-ethnique afin de déterminer de quelles étaient leurs origines. Dans un second temps les auteurs ont tenté de faire concorder ces *admixture events* à des bouleversement historiques connus (grandes invasions, migrations ou colonisations).

Pour notre projet, nous nous somme inspirés de cet article pour mettre en évidence des liens entre proximité géographique (resp. éloignement géographique) de sujets et similarités (resp. dissimilarités) dans leur génome.

Après quelques rappels biologique, nous présenterons les méthodes et outils statistiques que nous avons utilisés pour mener à bien notre recherche. Enfin, dans une troisième partie, nous décrirons et commenterons les résultats obtenus.

2 Rappels de génétique

2.1 Structure de l'information génétique

Toute l'information génétique de l'Homme, son génome, est contenue dans l'acide désoxyribonucléïque (ADN). Ce matériel génétique contient sous forme codée toutes les informations relatives au développement et au maintien des organismes vivants. L'ADN a une stucture particulière dite *en double hélice* : deux brins composés d'une succession de nucléotides accrochés les uns aux autres par des liaisons phosphates. Chaque brin de l'ADN est constitué d'une suite d'un des quatre nucléotides, que sont l'adénosine, la cytosine, la guanine et la thymine.

[2]

1. Hellenthal, Garrett, et al. "A genetic atlas of human admixture history." Science 343.6172 (2014) : 747-751.

2. https ://fr.wikipedia.org

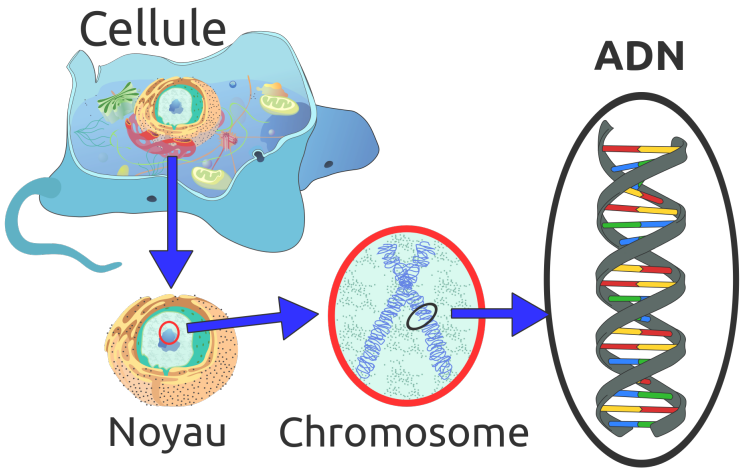

FIGURE 1 – Schéma de l'Acide DésoxyriboNucléique

2.2 Polymorphismes génétiquet et *Single Nucleotide Polymorphism*

Le gène correspond à un court fragment ADN et constitue le support de l'information génétique. On appelle génome l'ensemble des gènes. Ces gènes sont alors transcrits en acides ribonucléiques (ARN) qui seront par la suite traduits en protéines.

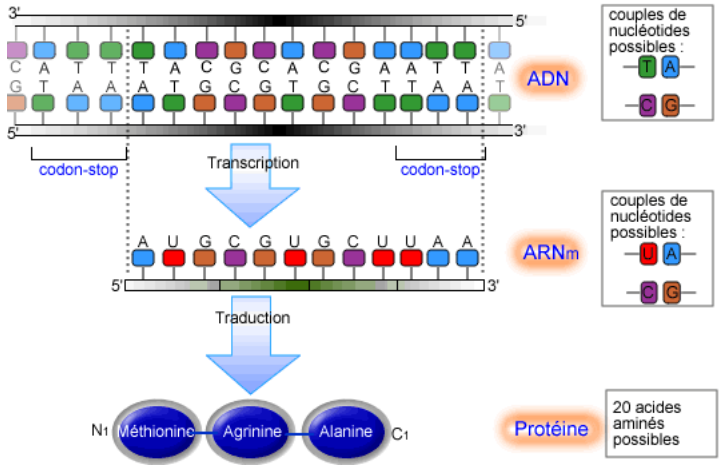

FIGURE 2 – Etapes de la synthèse d'une protéine

[3]

L'intégrité du génome dans son enchaînement de bases nucléotidique est primordiale. Une altération du génome peut avoir des conséquences sur les gènes produits et donc avoir un impact sur les fonctions de l'organisme. Un polymorphisme génétique désigne la coexistence de plusieurs allèles pour un gène ou locus donné.

Il existe plusieurs types de polymorphismes génétiques :

— Séquences répétées en tandem (microsatellite)
— Insertion-délétion
— Polymorphisme du nombre de copies (CNV)
— Polymorphismes mono-nucléotidique (SNP)

Le type de polymorphisme qui nous intéresse dans ce projet est le *Single Nucleotide Polymorphism* (SNP). Les SNP sont la plus petite forme de polymorphisme existante car elles n'affectent qu'une seule paire de bases. Elles constituent près de 90 % des polymorphismes répertoriés.

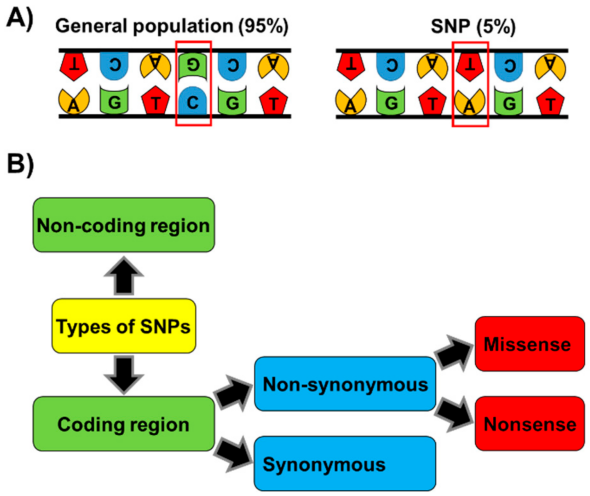

FIGURE 3 – Types de SNP
[4]

3 Présentation de l'étude et des données

La base de données étudiée a été obtenu par génotypage SNP sur puce SNP (puce Illumina Human660W-Quad v1.0 BeadChip). Les prélévements ont été effectués sur 158 individus dont les origines sont Eurasiennes. On observe alors la nature de plus de 600.000 SNPs sur l'ensemble des individus.

On dispose également d'informations autre que génétique sur les individus, comme leur sexe, leur nationalité, la nature du prélévement (salive ou sang). Au total, des individus de 11 nationalités différentes font partie de l'étude.

A partir de ces données génétiques et grâce à des méthodes statistiques appropriées, l'article associé à ces données cherchait à produire un atlas de l'histoire mondiale des mélanges humains ainsi qu'à identifier des dates et événements clefs de l'histoire humaine.

L'article initial part de l'observation que les données génétiques modernes combinées à des méthodes statistiques appropriées ont le potentiel de contribuer considérablement à notre compréhension de l'histoire humaine. Les auteurs se servent

de la structure génomique de populations mélangées afin d'essayer de retraçer l'histoire de ces populations et déviner certains évenements. Ils ont alors produit un atlas de l'histoire mondiale des mélanges humains, construit en utilisant uniquement des données génétiques et englobant plus de 100 événements survenus au cours des 4000 dernières années.

Nous avons utilisé les données SNP de 158 individus originaires de 11 nationalités différentes, et nous avons décidé de chercher à mettre en avant une proximité (ou non) au niveau des SNPs suivant l'origine des individus de l'étude. Pour cela avons utilisé différentes méthodes, de la réduction de données avec une PCA, de la classifications avec des algorithmes de type Logistic Regression ainsi que Random Forest, et enfin une classification non supervisée afin de voir si les clusters trouvés pourraient correspondre à nos différentes populations.

4 Méthodologie

Durant ce projet nous avons tenté plusieurs approches afin de domprendre et d'exploiter les données.

1. **Données manquantes**
   Notre premier réflexe a été de rechercher les SNP pour lesquels il y avait des données manquantes et nous les avons supprimées. Cette action a réduit notre base de données à 497899 SNP.
2. **Analyses en Composantes Principales**
   Dans un second temps nous avons effectué une Analyse en composante Principale sur les données sans données manquantes. Le but était d'essayer de retrouver les 11 nationalités et projetant les individus dans le plan principal.

   Pour aller plus loin nous avons choisis des paires de nationalités de manière a créer des paires géographiquement proches et géographiquement distinctes. Nous avons ensuite refait une analyse en composante principales sur ces données afin d'observer si l'algorithme était capable de différencier les deux nationalités uniquement en se basant sur l'information issue des SNP.
3. **Algorithme des kmeans**
   Nous avons ensuite fait tourner un algorithme des k-means en choisissant la construction de 11 groupes pour tenter de retrouver les 11 nationalités du jeu de données et d'établir une comparaison avec la projection dans le plan principale cité précédement des 158 individus de l'étude.
4. **Regression Logistique Multiclasse et Fôrets Aléatoires**
   Enfin, nous avons entrainé deux algorithmes de Machine Learning à prédire l'appartenance à une nationalité en se basant uniquement sur l'information contenue dans les SNP : Une régression Logisique Multiclasse (pour les 11

nationnalités) et un Random Forest. Pour permettre l'apprentissage des algorithmes nous avons séparé les données en un jeu d'entrainement et un jeu de test avec une proportion de respectivement 2/3 et 1/3.

5 Analyse et Résultats

Après avoir nétoyer le jeu de données des SNP non observés, ce dernier est constitué de 497.899 SNP et de 158 individus issus de 11 nationnalités (Figure 4) :

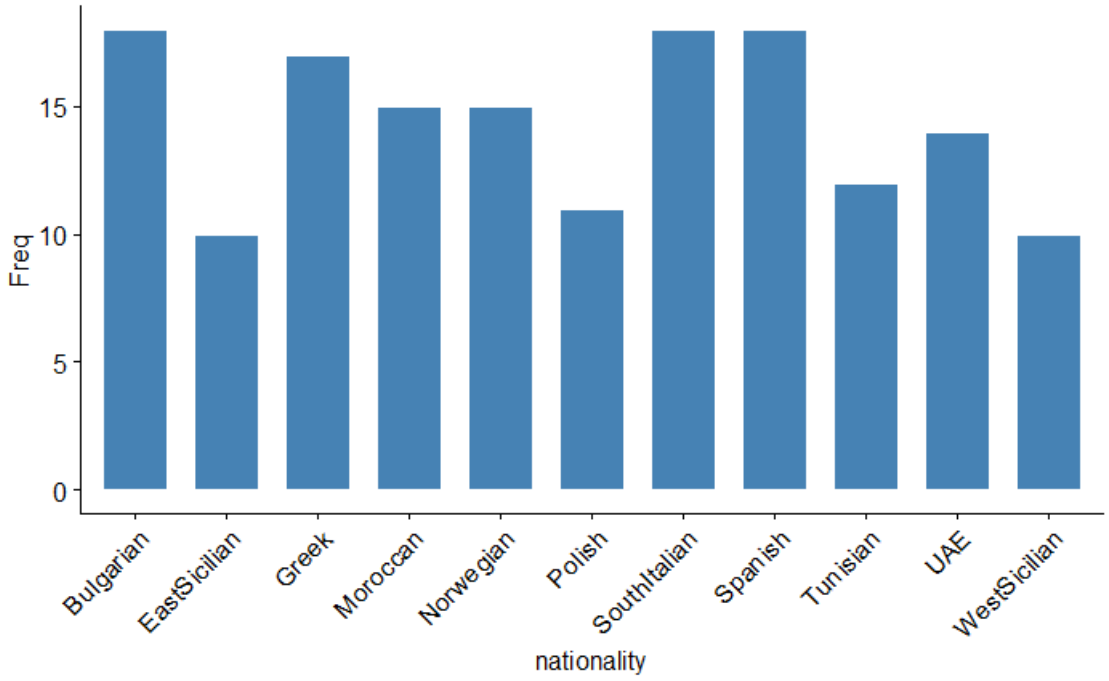

FIGURE 4 – Répartition des individus de l'étude dans les différentes populations

5.1 Analyse en Composantes Principales

A travers différentes ACP, on cherche à savoir si il existe un lien entre proximité (resp. éloignement) géographique et similitude (resp. dissimilitudes) au niveau des SNP observés.

ACP sur l'ensemble des données

A travers une autre approche, nous nous sommes intéressés aux résultats obtenus par une ACP appliqué à l'ensemble des données.

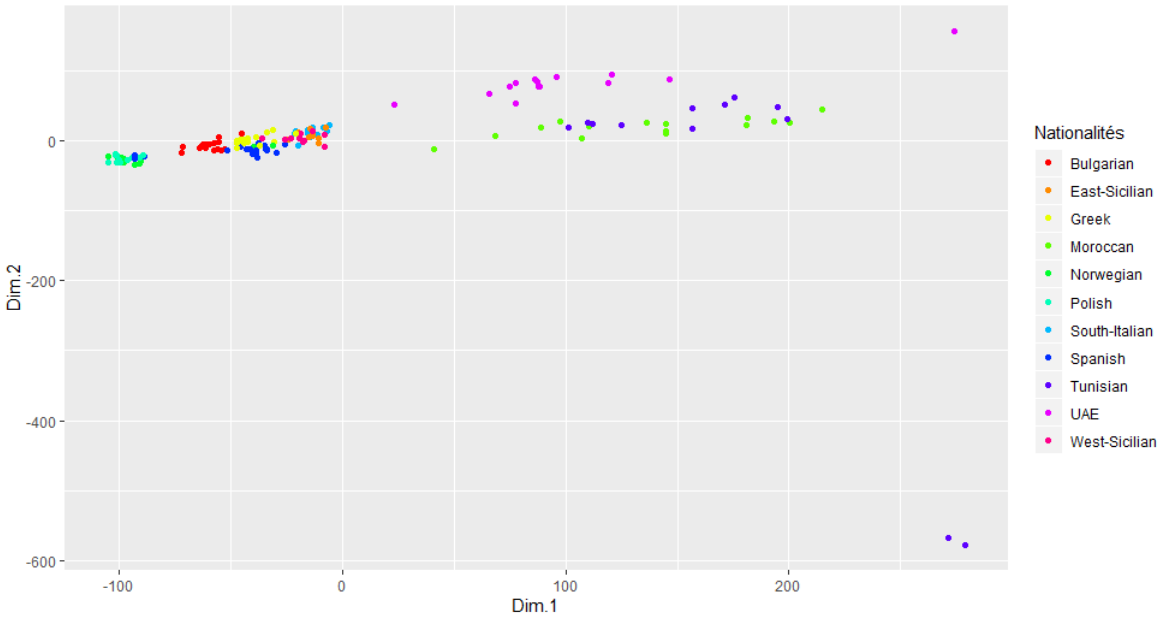

FIGURE 5 – Analyse en composantes principales sur la population totale

On remarque que l'ACP a du mal à distinguer les groupes East-Sicilian, West-Sicilian e South-Italian. Ce résultat va dans le sens de notre hypothèse de travail à savoir : les pays proches géographiquement on une population dont le génome est peu différencié. On remarque que les Bulgare forment un groupe assez compact et distinct des autres tandis que les sujets issus des Emirats Arabes Unis forment un groupe distnic mais moins compact. Un seul groupe se discerne vraiment du reste des populations.

Individus géographiquement éloignés

Dans un premier temps, on s'est intéressé à deux groupes de populations d'origine géographiquement éloignés, à savoir *Moroccan* et *Norwegian*.

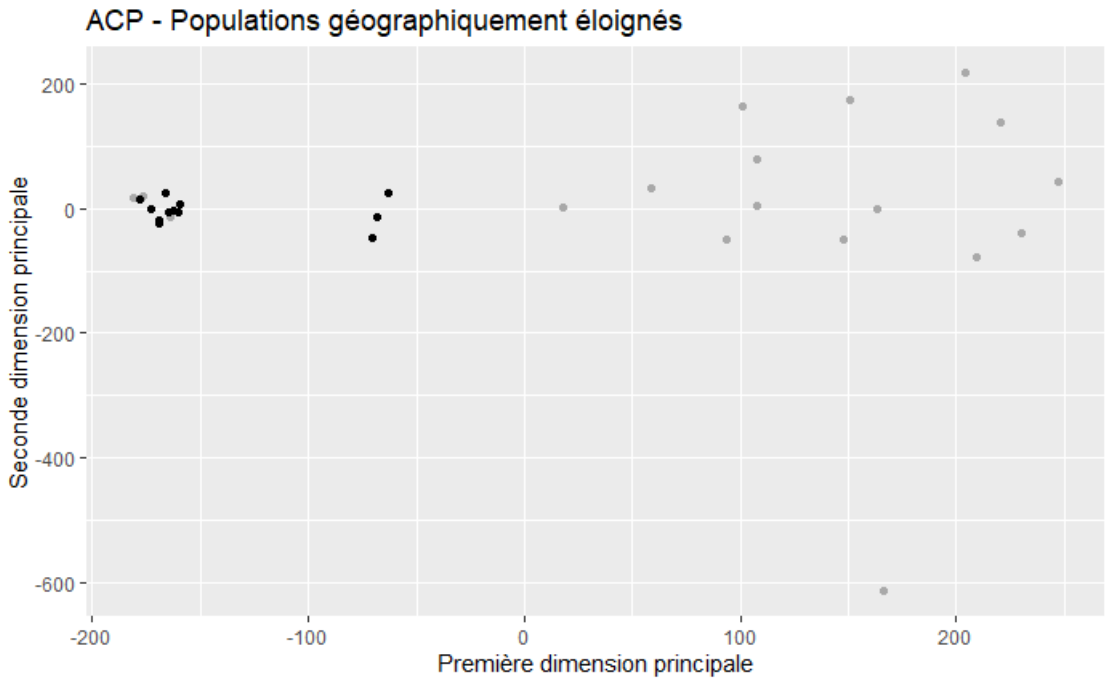

FIGURE 6 – Analyse en composantes principales : *Moroccan* vs *Norwegian*

La première ACP (Figure 6) nous permet d'observer une distinction des groupes dans ces nouvelles dimensions. Même si certains individus sont présent dans le groupe de gauche, un cission notable se créée entre les deux groupes. On peut alors penser qu'il existe une différence nette en terme de SNP entre ces deux populations.

De plus (Figure 7), en s'intéressant à la contribution de chaque SNP dans la décision de ces nouvelles dimensions. On oberve qu'un nombre restraint de SNP à une influence beaucoup plus importante que le reste des autres. Autrement dit, certaines formes de SNP sont plus spécifiques à une population qu'a une autre.

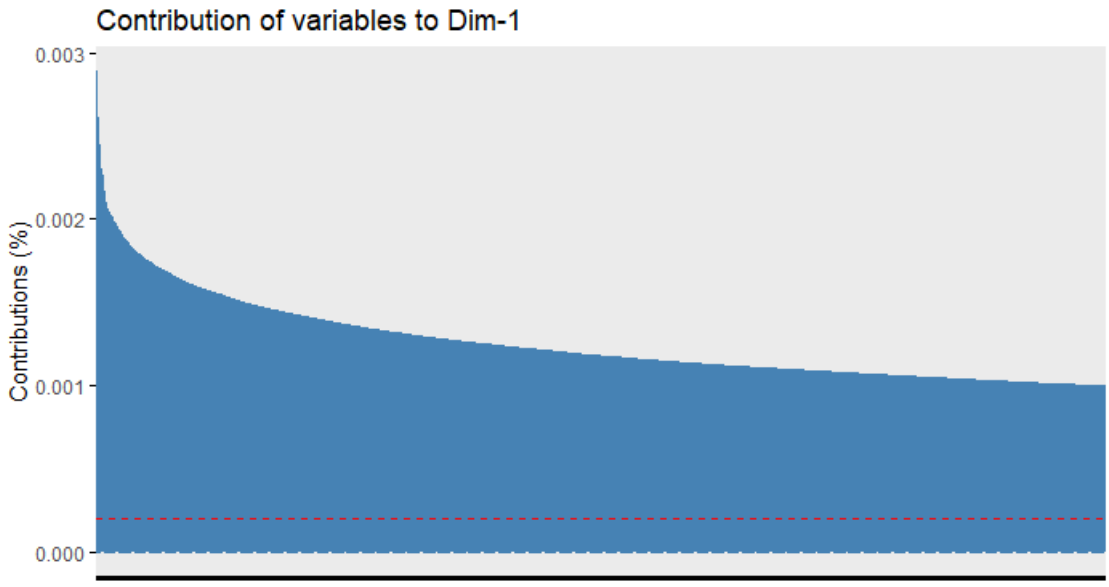

FIGURE 7 – Contribution des SNP dans la 1ere composante principale : *Moroccan* vs *Norwegian*

Individus géographiquement proches

Dans un second temps, on s'est intéressé à deux groupes de populations d'origine géographiquement proches, à savoir *EastSicilian* et *WestSicilian*.

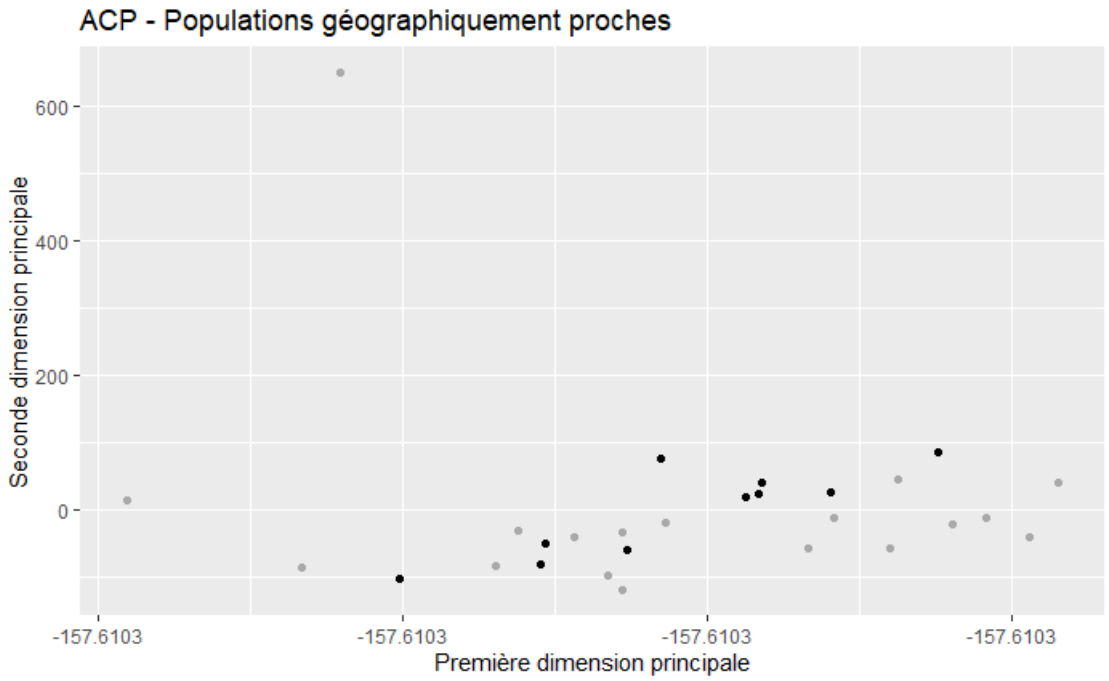

FIGURE 8 – Analyse en composantes principales : *EastSicilian* vs *WestSicilian*

La seconde ACP (Figure 8) ne nous permet pas d'observer un réelle distinction des groupes dans ces nouvelles dimensions. De plus (Figure 9, lorsque l'on s'interesse à la contribution des SNP dans la décision des dimensions, on observe que tous les SNP ont une influence identique dans ce choix.

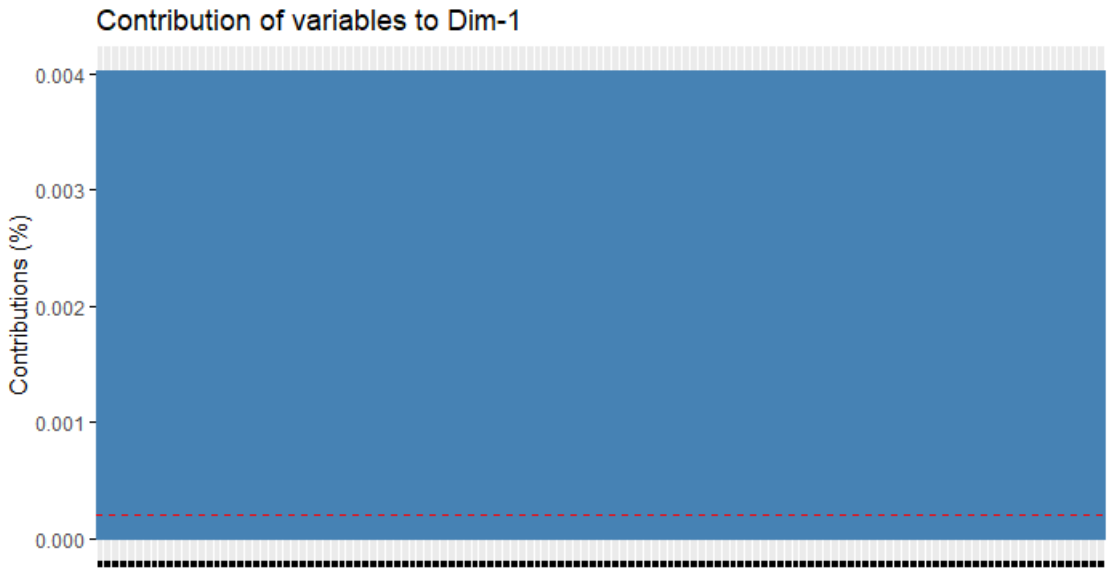

FIGURE 9 – Contribution des SNP dans la 1ere composante principale : *EastSicilian* vs *WestSicilian*

Alors, contrairement au premier cas, on peut ici penser que la proximité géographique des deux populations étudiées ne permet pas de déceler des différences significatives en terme de contenu génétique.

5.2 Clustering

Dans cette partie nous présentons la classification (non superviée) en 11 groupe des données obtenue à partir d'un k-means.

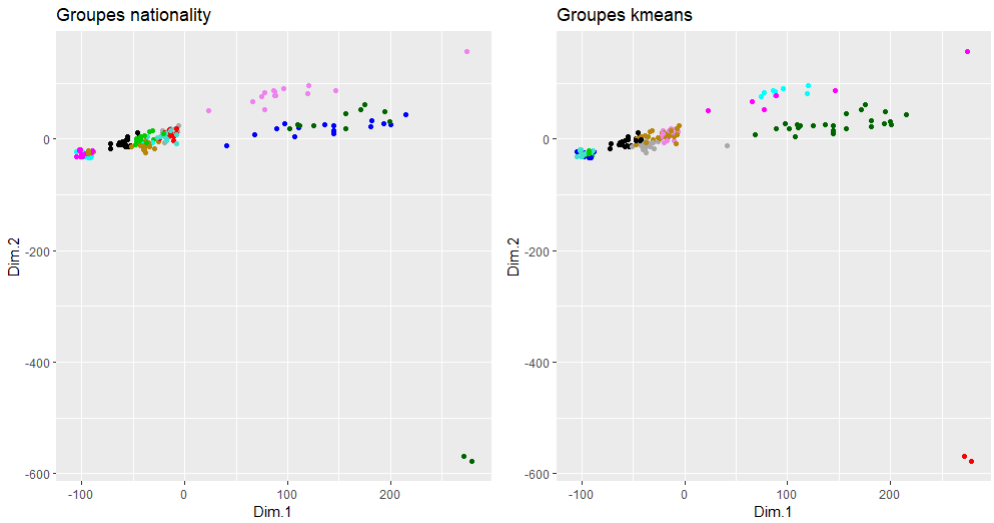

FIGURE 10 – Comparaison des vraies nationalités avec le clustering des k-means

Les différents clusters choisis par l'algorithme ne coïncident pas avec les réels groupes de nationalités des individus. Cette erreur est nottament due à certaines valeurs extremes (observables en bas a gauche) qui sont considérées comme une nationalité à part entière.

5.3 Régression Logistique et Random Forest

Dans un premier temps nous avons de classifier les populations en utilisant toutes les données. Puis, dans un second temps, nous nous sommes concentrés sur deux binomes, l'un avec deux populations proches géographiquement, et l'autre avec deux populations éloignées géographiquement.

Nous commençons avec la régression logistique, étant donné que nous avons 11 nationalités différentes nous utilisons une classification dite 'balanced' afin que l'on ai pas de nationalité mise de côté. Nous obtenons un score de 0.32 sur le jeu de test.

Une fois la PCA appliquée, nous réessayons en essayant de garder le maximum possible d'informations c'est à dire ici 150 composants pour la PCA. Le résultat alors obtenu est de 0.35, ce qui reste très en dessous de ce que nous recherchons.

Nous avons alors décidé de passer sur un algorithme de Random Forest, qui semble à priori plus adapté au type de données que nous avons ici. Nous utilisons une forêt de 80 arbres de décisions, et nous obtenons un score de 0.075, ce qui est en dessous d'une simple fonction aléatoire. Néanmoins une forêt de 80 arbres est très petite alors que nous avons sans PCA près d'un demi million de variables explicatives.

Afin d'y remédier, nous réappliquons la PCA, toujours avec 150 composants et une forêt de 80 arbres. Le résultat obtenu est alors de 0.62, ce qui est alors beaucoup plus intéressant même si ça n'est toujours pas suffisant.

A présent nous allons nous focaliser sur deux populations éloignées géographiquement et tenter de les classifier plus efficacement. Nous choisissons les individus norvégiens ainsi que les marocains.

Nous utilisons toujours les mêmes algorithmes avec une même paramétrisation, ainsi qu'une PCA de 30 composants. Nous obtenons alors un score de 0.8 concenant la Régression Logistique et un score parfait de 1 avec la Random Forest. Au vue des résultats il semble évident que l'algorithme a bien trouvé des différences notables dans les SNPs de ces populations éloignées.

Enfin, nous réessayons la même chose avec deux populations très proches géographiquement, les siciliens de l'ouest et ceux de l'est.

La PCA est donc choisie avec 20 composants et les résultats obtenus sont de 0.57 avec les deux algorithmes. Montrant à l'inverse du cas précedant qu'il y a une différence beaucoup moins grande entre ces deux populations au niveau des SNPs.

6 Conclusion

A travers ce projet nous avons mis en évidence plusieurs points.

Dans un premier temps, l'Analyse en Composantes Principales a permis de mettre en évidence que la proximité géographique justifie en effet d'une proximité dans l'information génétique.

Dans un second temps, les algorithmes de machine learning (Logistic Regression et Random Forest) ont montrés qu'il était possible de construire un "détecteur" de nationalité en se basant simplement sur l'information génétique des sujets. Cependant il est parfoit difficile d'obtenir des résultats satisfaisant car l'information génétique de certains individus est parfois très proche si leurs nationalités sont "voisines".

Des questions se posent alors : pourquoi certains sujets ont une proximité génétique très différentes des autres individus avec la même nationalité. Et, de plus, pourquoi certains pays voisins tels que la Tunisie et le Maroc n'ont pas une proximité aussi grande que la Grèce et l'Espagne ?

Une réponse possible se trouve dans l'article initial, qui réussi à jusifier une proximité génétique au travers du prisme de des mouvements de populations (colonialisme, guerres etc...). Les auteurs parviennent également à retrouver une estimation de la date historique à laquelle cela s'est produit l'événement en question puisque, plus l'évenement est lointain, plus l'information génétique lié à cette période est présent en petite quantité.

Une ouverture intéressante à ce projet serait la mise en évidence des SNP responsables de la reconnaissance de chaque nationnalité. Une fois les SNP listés, des tests statistiques comme le χ^2 de Pearson pourraient montrer (ou infirmer) une plus grande présence de certains formes de SNP au sein d'une certaine nationalité.

Pour conclure, il est intéressant d'observer que le patrimoine de la race humaine, au sens historique, est inscrit dans son génome et que c'est précisement les mélanges ethniques dûs aux migrations qui en font la richesse.

7 References

- Hellenthal, G., Busby, G. B., Band, G., Wilson, J. F., Capelli, C., Falush, D., & Myers, S. (2014). A genetic atlas of human admixture history. Science, 343(6172), 747-751.
- Taing, L. (2012). Approches bioinformatiques pour l'exploitation des données génomiques (Doctoral dissertation).
- https://botany.natur.cuni.cz/hodnocenidat/Lesson_05_tutorial.pdf
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4209567/
- http://admixturemap.paintmychromosomes.com/
- https://www.well.ox.ac.uk/~gav/admixture/2014-science-final/resources/FAQ.pdf

8 Annexes

Les algorithmes de machine Learning on été implémentée en python, toutes les autres analyses on été effectuées sur R.

Lecture des données

```
data = read.table("GSE53626_series_matrix2.txt", sep='\t', header=T, comment.
char="!", row.names=1, na.string="NC")

nationality = as.factor(t(read.table("GSE53626_series_matrix2.txt", skip=42,
nrows=1))[-1])
table(nationality)

length(nationality)

## [1] 158

library(stringr)
natio = as.factor(unlist(lapply(nationality, str_sub, 14)))
table(natio)

## natio
##    Bulgarian EastSicilian        Greek     Moroccan    Norwegian
##           18           10           17           15           15
##       Polish SouthItalian      Spanish     Tunisian          UAE
##           11           18           18           12           14
## WestSicilian
##           10

sexe = as.factor(t(read.table("GSE53626_series_matrix2.txt", skip=41, nrows=1
))[-1])
table(sexe)

## sexe
## gender: female   gender: male
##              4            154

length(sexe)

## [1] 158
```

Gestion des données manquantes

```
data2 = data[rowSums(is.na(data)) == 0, ]
```

ACP

```
library(FactoMineR)

res.pca = PCA(X = t(data2))

library(factoextra)

## Loading required package: ggplot2
## Welcome! Want to learn more? See two factoextra-related books at https://g
oo.gl/ve3WBa

ind = get_pca_ind(res.pca)

data_ind = ind$coord[,1:2]
data_ind = cbind(data_ind, natio)

library(ggplot2)

data_ind = as.data.frame(data_ind)
ggplot(data_ind, aes(x = data_ind[,1],y = data_ind[,2], colour = factor(data_
ind[,3]))) + geom_point() +  scale_color_manual(name="Nationalités", labels =
c("Bulgarian", "East-Sicilian", "Greek", "Moroccan", "Norwegian", "Polish", "
South-Italian", "Spanish", "Tunisian", "UAE", "West-Sicilian"), values = rain
bow(11)) + xlab("Dim.1") + ylab("Dim.2")


var <- get_pca_var(res.pca)

# Contributions des variables à PC1
fviz_contrib(res.pca, choice = "var", axes = 1, top = 500)


# Contributions des variables à PC2
fviz_contrib(res.pca, choice = "var", axes = 2, top = 500)
```

Clustering: k-means

```
res.kmeans = kmeans(x = t(data2), centers = 11)

plot_grid(
ggplot(data_ind, aes(x = data_ind[,1],y = data_ind[,2])) + geom_point(colour
= factor(data_ind[,3])) + xlab("Dim.1") + ylab("Dim.2") + ggtitle("Groupes na
tionality"),
ggplot(data_ind, aes(x = data_ind[,1],y = data_ind[,2])) + geom_point(colour
= factor(res.kmeans$cluster))  + xlab("Dim.1") + ylab("Dim.2") + ggtitle("Gro
upes kmeans"))
```

In [1]:
```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [2]:
```
df = pd.read_table("D:/Utilisateurs/Samuel/Bureau/Histoire_pop/GSE53626_series_matrix.txt.gz", sep='\t', comment="!", index_c
ol='ID_REF')
print(df.shape)
df.head()
```

```
(657366, 158)
```

Out[2]:

ID_REF	GSM1297335	GSM1297336	GSM1297337	GSM1297338	GSM1297339	GSM1297340	GSM1297341	GSM1297342	GSM1297343	GSM1297344	…	GSI
200003	AA	AB	AB	AB	BB	AB	AA	AA	AB	AB	…	
200006	AB	AB	AB	AB	AA	AB	BB	BB	AB	AB	…	
200047	AA	AA	AA	AA	AA	AA	AB	AA	AA	AA	…	
200050	BB	BB	BB	BB	BB	BB	AB	BB	BB	BB	…	
200052	BB	BB	BB	BB	BB	BB	AB	BB	BB	BB	…	

5 rows × 158 columns

In [3]:
```
df2 = df.reset_index(drop=True)
df2 = df2.replace("AB",0).replace("AA",1).replace("BB",2).replace('NC',np.nan)
df2 = df2.dropna(axis=0, how='any')
df2.head()
```

Out[3]:

	GSM1297335	GSM1297336	GSM1297337	GSM1297338	GSM1297339	GSM1297340	GSM1297341	GSM1297342	GSM1297343	GSM1297344	…	GSM1297
0	1.0	0.0	0.0	0.0	2.0	0.0	1.0	1.0	0.0	0.0	…	
1	0.0	0.0	0.0	0.0	1.0	0.0	2.0	2.0	0.0	0.0	…	
3	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	2.0	2.0	…	
4	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	2.0	2.0	…	
5	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	…	

5 rows × 158 columns

In [4]:
```
nationalite = pd.read_csv('nationalite.csv', header = None)
nationalite[0] = nationalite[0].apply(lambda x : x[:-1])

nationalite.shape
```

Out[4]:
```
(158, 1)
```

In [5]:
```
X_train, X_test, y_train, y_test = train_test_split(np.transpose(df2), nationalite[0], test_size=0.33, random_state=42)
```

In [6]:
```
from sklearn.linear_model import LogisticRegression

# Training Logistic Regression
classifier = LogisticRegression(class_weight='balanced', multi_class='ovr', solver='liblinear')
classifier.fit(X_train , y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 0.32075471698113206
```

In [7]:
```
from sklearn.ensemble import RandomForestClassifier

# Training Random Forest
clf = RandomForestClassifier(n_estimators=80, random_state=42)
clf.fit(X_train , y_train)

# Predicting the Test set results
y_pred2 = clf.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred2))
```

```
Le score est: 0.07547169811320754
```

In [8]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=150)
df3 = pca.fit_transform(np.transpose(df2))

print(pca.explained_variance_ratio_)
```

```
[0.01077863 0.00887085 0.00772295 0.00738386 0.00705369 0.00691664
 0.00681656 0.00678369 0.00677638 0.00675719 0.00674393 0.00672777
 0.00672118 0.0067048  0.00669721 0.0066875  0.00667657 0.00667428
 0.00666512 0.00664526 0.00664305 0.00664042 0.00663534 0.00662194
 0.00661958 0.00661518 0.00660387 0.00660123 0.0065959  0.00659325
 0.00658507 0.00658292 0.00657207 0.00657001 0.00656282 0.00655486
 0.00654881 0.00654426 0.00653716 0.00653079 0.0065211  0.00651985
 0.00651459 0.00650749 0.00650531 0.00649642 0.00649361 0.00648771
 0.00648138 0.00647504 0.00646926 0.00646377 0.00644852 0.00644402
 0.00644196 0.00643478 0.00643082 0.00642863 0.0064252  0.00642023
 0.00641325 0.0064054  0.00640108 0.00639562 0.00639243 0.00638865
 0.00638507 0.00637703 0.00637388 0.00636897 0.00636772 0.00635798
 0.00635489 0.00634982 0.00634596 0.00634167 0.00633322 0.00632709
 0.00632142 0.00631828 0.00631361 0.00630935 0.00630254 0.00629601
 0.00629117 0.00628515 0.00628406 0.00627788 0.00627264 0.00627026
 0.00626831 0.00626392 0.00625207 0.00625084 0.0062495  0.0062389
 0.00623849 0.00623052 0.00622209 0.00621692 0.00621605 0.00621386
 0.00620496 0.00619742 0.00619145 0.00618998 0.00618935 0.00618293
 0.00617504 0.00617103 0.00616518 0.00616339 0.00615614 0.00615089
 0.00614646 0.00613709 0.00613185 0.00612997 0.00612706 0.00611786
 0.00611242 0.00610569 0.00609887 0.00609259 0.00608671 0.00608138
 0.0060775  0.00607312 0.00606708 0.00606352 0.00605374 0.00605002
 0.00604508 0.00603961 0.00603059 0.00602463 0.00601895 0.0060144
 0.00600814 0.00600187 0.00599152 0.00598896 0.00597893 0.00596697
 0.00596007 0.00595574 0.00594672 0.00594501 0.00593617 0.00592768]
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(df3, nationalite[0], test_size=0.33, random_state=42)
```

In [10]:
```
# Training Logistic Regression
classifier = LogisticRegression(class_weight='balanced', multi_class='ovr', solver='liblinear')
classifier.fit(X_train , y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 0.3584905660377358
```

In [11]:
```
# Training Random Forest
clf = RandomForestClassifier(n_estimators=80, random_state=42)
clf.fit(X_train , y_train)

# Predicting the Test set results
y_pred = clf.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 0.6226415094339622
```

In [32]:
```
norvege_maroc = df2.iloc[:,(nationalite[nationalite[0].isin(['Moroccan', 'Norwegian'])].index)]

pca = PCA(n_components=30)
norvege_maroc_pca = pca.fit_transform(np.transpose(norvege_maroc))

print(pca.explained_variance_ratio_)
```

```
[4.34489950e-02 3.53895169e-02 3.53545110e-02 3.51613466e-02
 3.49819662e-02 3.49268946e-02 3.48405730e-02 3.47620884e-02
 3.47387154e-02 3.47013225e-02 3.46247975e-02 3.45490150e-02
 3.44553722e-02 3.44211115e-02 3.41764821e-02 3.41172658e-02
 3.39940621e-02 3.39377261e-02 3.39067845e-02 3.37354558e-02
 3.36892107e-02 3.35311855e-02 3.35121655e-02 3.34492306e-02
 3.33537547e-02 3.32462425e-02 3.31493222e-02 3.30684635e-02
 3.27762927e-02 1.91839306e-28]
```

In [42]:
```
X_train, X_test, y_train, y_test = train_test_split(norvege_maroc_pca, nationalite[nationalite[0].isin(['Moroccan', 'Norwegia
n'])][0], test_size=0.33, random_state=42)
```

In [43]:
```
# Training Logistic Regression
classifier = LogisticRegression(class_weight='balanced', multi_class='ovr', solver='liblinear')
classifier.fit(X_train , y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 0.8
```

In [44]:
```
# Training Random Forest
clf = RandomForestClassifier(n_estimators=80, random_state=42)
clf.fit(X_train , y_train)

# Predicting the Test set results
y_pred = clf.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 1.0
```

In [53]:
```
sicilian = df2.iloc[:,(nationalite[nationalite[0].isin(['EastSicilian', 'WestSicilian'])].index)]

pca = PCA(n_components=20)
sicilian_pca = pca.fit_transform(np.transpose(sicilian))

print(pca.explained_variance_ratio_)
```

```
[5.39757027e-02 5.37282225e-02 5.35832942e-02 5.35185801e-02
 5.32304630e-02 5.30733565e-02 5.29771118e-02 5.29392157e-02
 5.26853130e-02 5.26343175e-02 5.24242077e-02 5.23143884e-02
 5.22151678e-02 5.21415122e-02 5.20656202e-02 5.19021936e-02
 5.17340182e-02 5.16562375e-02 5.12010773e-02 6.44443151e-27]
```

In [54]:
```
X_train, X_test, y_train, y_test = train_test_split(sicilian_pca, nationalite[nationalite[0].isin(['EastSicilian', 'WestSicil
ian'])][0], test_size=0.33, random_state=42)
```

In [55]:
```
# Training Logistic Regression
classifier = LogisticRegression(class_weight='balanced', multi_class='ovr', solver='liblinear')
classifier.fit(X_train , y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 0.5714285714285714
```

In [56]:
```
# Training Random Forest
clf = RandomForestClassifier(n_estimators=80, random_state=42)
clf.fit(X_train , y_train)

# Predicting the Test set results
y_pred = clf.predict(X_test)

# Scoring
print('Le score est:',accuracy_score(y_test , y_pred))
```

```
Le score est: 0.5714285714285714
```