TP : Diversité génétique d'une population structurée

On parle de population structurée dès lors que l'on s'écarte des hypthèses de panmixie du modèle de Wright-Fisher, autrement dit dès lors que les individus de la population ne sont pas tous interchangeables mais appartiennent à différents groupes distinguables les uns des autres. Nous allons voir ici comment simuler des données génétiques dans un modèle structuré, en prenant l'exemple d'un modèle dans lequel plusieurs populations ont divergé d'une population ancestrale commune (il existe de nombreux autres modèles de structure). A partir de cet exemple, nous verrons comment visualiser la structure dans des données génétiques et comment la quantifier à l'aide d'un indice standard appelé Fst.

1. Simulation de données génétiques sous un modèle de divergence

On considère ici deux populations ayant divergé d'une population ancestrale commune et n'ayant pas échangé de migrants depuis. Les deux populations ainsi que la population ancestrale ont toutes la même taille (N haploïdes). Supposons par exemple qu'on échantillone 10 individus dans chaque population, que la divergence a eu lieu il y a 20,000 générations, et gardons par ailleurs les mêmes paramètres qu'au TP précédent. Pour simuler ce scénario avec *coala*, on utilise la commande:

```
library(coala)
activate_ms(priority = 500)
loc=1000
L=1000
N=10000
mu=2*10^(-8)
theta=2*N*mu
t=20000/(2*N) # L'unité de temps pour ms est 2N générations
structModel <- coal_model(sample_size=c(10, 10), loci_number=loc, loci_length=L,ploidy=1) +
feat_mutation(rate=theta*L) + feat_pop_merge(t,2,1) + sumstat_seg_sites() + sumstat_trees()
structRes <- simulate(structModel)
```

à laquelle on peut ajouter, comme dans les TP précédents, d'autres arguments de type *sumstat* en fonction du type de statistiques que l'on veut générer. Dans cette commande, l'argument *sample_size=c(10, 10)* indique qu'on considère deux populations avec 10 séquences éhantillonées dans chacune, et la commande *feat_pop_merge(t,2,1)* indique que la population 2 fusionne avec la population 1 à un temps t dans le passé.

On peut visualiser la généalogie des séquences au locus 1 à l'aide de la commande suivante

```
library(phyclust)
tree=read.tree(text=structRes$trees[[1]])
plot(tree,show.tip.label=TRUE,no.margin=FALSE,direction="downwards")
```

Répétez l'opération en regardant d'autres locus. Qu'observe-t-on?

Observe-t-on la méme chose pour un temps de divergence de 5,000 générations?

2. Visualisation de la structure dans les données

Un moyen classique pour visualiser la structure des données génétiques consiste à effectuer une Analyse en Composantes Principales (ACP) de la matrice représentant les allèles (0 ou 1) portés par chaque séquence à chaque SNP. On obtient cette matrice par les commandes

```
seq.mat=as.matrix(structRes2$seg_sites[[1]])
for (i in 2:loc){
    seq.mat=cbind(seq.mat,as.matrix(structRes2$seg_sites[[i]]))
}
```

L'ACP de cette matrice peut s'effectuer (par exemple) à l'aide des commandes suivantes

```
acp=prcomp(seq.mat,scale=F)
plot(acp$x[,1],acp$x[,2],xlab='PC1',ylab='PC2',type='n')
text(acp$x[,1],acp$x[,2],labels=1:20)
```

On voit que la différentiation entre populations est beaucoup plus claire ici que sur les arbres tracés précédemment. Comment l'expliquez vous?

Recommencez l'analyse pour un modèle avec 3 populations, deux (numéros 1 et 2) de divergence récente et une (numéro 3) de divergence plus lointaine. Le résultat obtenu vous paraît-il logique? Essayez aussi pour comparer avec un modèle de population panmictique.

L'utilisation de l'ACP pour visualiser la structure génétique d'un échantillon de séquences est devenue très classique en génétique des populations. Dans le cas d'espèces diploïdes, celle-ci peut s'appliquer à des tableaux de 0/1/2 (au lieu de 0/1) représentant le nombre d'allèles alternatifs pour un individu et un SNP donné. Notez aussi que de nombreux outils plus spécifiques ont été développés pour analyser la structure d'un échantillon de données génétiques (nombre de groupes distincts, taux d'hybridation ...), le premier (et le plus connu) d'entre eux étant le logiciel Structure (Pritchard et al, 2000).

3. Définition du Fst

Considérons un échantillon de séquences dont on sait a priori (ou suite à une ACP) qu'il est structuré en un certain nombre de populations. Le Fst est un indice visant à décrire la proportion de la diversité génétique qui est due à cette structure. Autrement dit, il doit valoir 0 si la population n'est en fait pas structurée. A l'opposé, sa valeur maximale (aucune diversité intra groupes) est par construction de 1.

La définition du Fst fait appel à la notion d'IBS (Identity by State, en anglais). Deux séquences sont dites IBS pour un locus donné si elles portent exactement la même suite d'allèles à ce locus (ou simplement le même allèle, 0 ou 1, si le locus est un SNP). Sous les hypothèses de l'Infinite Site Model utilisé classiquement en coalescence, cela veut dire qu'aucune mutation n'est apparue depuis leur ancêtre commun le plus récent.

La définition la plus classique du Fst est

$$Fst = \frac{f_0 - \bar{f}}{1 - \bar{f}}$$

o'u f_0 est la probabilité que deux séquences issues de la même sous-population soient IBS et $\bar{f}$ est la probabilité que deux séquences choisies au hassard dans l'ensemble de la population soient IBS. De manière équivalente, on peut définit le Fst en fonction de l'hétérozygotie, qui est la probabilité que deux séquences choisies au hasard soient distinctes. On écrit ainsi

$$Fst = \frac{H_T - H_S}{H_T}$$

o'u $H_T = 1 - \bar{f}$ est l'hétérozygotie dans l'échantillon total (T) et $H_S = 1 - f_0$ est l'hétérozygotie à l'intérieur d'une sous-population (S pour same). Notez que H_S est une moyenne sur les sous-populations, car elle peut différer de l'une à l'autre. Cette deuxième définition est peut être plus intuitive car elle fait apparaître clairement que le Fst va de 0 à 1, avec une valeur de 0 si $H_S = H_T$ (c'est à dire quand la notion de sous-population n'est pas pertinente) et 1 si $H_S = 0$.

Enfin, il est important de noter que le Fst est un indice théorique, qui découle d'un certain modèle d'évolution. Cette quantité théorique est à distinguer de l'estimation qui peut en être faite à partir de données réelles, qui est par définition imparfaite. Nous allons voir maintenant un moyen d'effectuer cette estimation.

4. Calcul du Fst

Compléter la fonction ci-dessous, qui permet de calculer la probabilité que pour un locus donné, deux haplotypes tirés au hasard dans une population soient identiques.

```
pIBS.haps <- function(snpData){
    # snpData : tableau de données donnant les allèles de n individus haploïdes (lignes)
    #   pour p positions polymorphes (colonnes)
    # samplesIDs : un vecteur avec les indices des séquences à considérer

    hapData <- apply(snpData, 1, paste, collapse="")
        # pour chaque individu, crée une chaîne de caractère donnant son haplotype
    hapTab <- table(hapData)
        # pour chaque haplotype, calcule le nombre d'individus portant cet haplotype
    # ....
}
```

Tester cette commande (en affichant éventuellement les résultats de calculs intermédiaires) pour les données d'un locus sans recombinaison simulé avec coala (répétez la simulation jusqu'à avoir au moin deux SNP) :

```
model=coal_model(sample_size=4,loci_number=1, loci_length=10000,ploidy=1) +
feat_mutation(rate=2) + sumstat_seg_sites()
simResults = simulate(model)
simResults$seg_sites[[1]]
pIBS.haps(simResults$seg_sites[[1]])
```

A l'aide de cette fonction, compléter la fonction suivante qui permet de calculer, à partir de l'attribut seg_sites d'un jeu de données simulé par coala, la probabilité IBS sur l'ensembe des locus, pour un sous ensemble d'individus donné:

```
pIBS.haps.coala <- function(segsites, samplesIDs){
    # segsites : liste de tableaux, telle que renvoyée par l'attribut seg_sites
    #   d'un résultat coala
    # samplesIDs : un vecteur avec les indices des individus à considérer

    nbSequences <- length(samplesIDs)
    nbSites <- length(segsites)
    ibs <- 0
    for(i in 1:nbSites){
            # ....
    }
    # ...
}
```

Tester cette commande pour la simulation coala suivante :

```
model=coal_model(sample_size=10,loci_number=100, loci_length=10000,ploidy=1) +
feat_mutation(rate=2) + sumstat_seg_sites()
simResults = simulate(model)
pIBS.haps.coala(simResults$seg_sites,1:5)
pIBS.haps.coala(simResults$seg_sites,6:10)
```

Pour finir, compléter la fonction ci-dessous qui calcule le Fst à partir de l'attribut seg_sites d'un jeu de données coala:

```
computeFst.haps.coala <- function(segsites, popList){
    # segsites : liste de tableaux, telle que renvoyée par l'attribut seg_sites
    #   d'un résultat coala
    # popList : liste de vecteurs donnant les indices des individus correspondant
    #   à chaque sous-population

    # probabilité IBS pour deux individus au hasard
    # ...
    # probabilité IBS pour deux individus au hasard dans la même sous-population
    # ...
    # calcul du Fst
    # ...
}
```

Tester cette fonction à l'aide de la simulation précédente, comme si les 5 premiers individus venaient d'une sous-population et les 5 derniers d'une autre. Le résultat est il logique?

```
computeFst.haps.coala(simResults$seg_sites, list(1:5,6:10))
```

Estimez maintenant le Fst pour le modèle considéré en début de tp, ou deux populations divergent il y a 5,000 ou 20,000 générations. Commentez le résultat.