Détection de régions évoluant sous sélection balancée

Carina Mugal, Anamaria Necsulea, Laure Ségurel

19 octobre 2023

1 Récupération des données et organisation de l'espace de travail

1. Connectez-vous à votre machine virtuelle. Dans votre dossier personnel `/home/ubuntu`, créez un répertoire nommé `Balancing_Selection` et utilisez-le comme répertoire courant.

```
cd /home/ubuntu

mkdir Balancing_Selection

cd Balancing_Selection
```

2. Pour cette partie, nous allons utiliser les mêmes fichiers VCF que pour la première partie du TP. Nous allons également utiliser les fichiers contenant l'information de la couverture par base. Nous allons donc copier ces fichiers dans ce nouveau répertoire.

```
cp /home/ubuntu/Variant_Calling/vcf/5-N00*_15ind_allsites_finalFiltered.vcf.gz .

cp /home/ubuntu/Variant_Calling/5-N00*_15ind_allsites_finalFiltered.ldepth.mean .
```

1.1 Prétraitement des fichiers

1. Nous allons d'abord filtrer notre fichier VCF pour enlever les sites pour lesquels nous n'avons pas l'information pour tous les 15 individus (sites où il y a des données manquantes). Nous allons faire ceci d'abord pour la séquence N00135.

```
vcftools --gzvcf 5-N00135_15ind_allsites_finalFiltered.vcf.gz --max-missing 1 \
--recode --recode-INFO-all --stdout > 6-N00135_15ind_allsites_finalFiltered_noMiss.vcf

bgzip 6-N00135_15ind_allsites_finalFiltered_noMiss.vcf
```

2. Nous allons ensuite filtrer ce fichier pour enlever les sites monomorphes (invariables).

```
vcftools --gzvcf 6-N00135_15ind_allsites_finalFiltered_noMiss.vcf.gz --maf 0.03 \
--recode --recode-INFO-all --stdout > 7-N00135_15ind_allsites_finalFiltered_noMiss_polym.vcf

bgzip 7-N00135_15ind_allsites_finalFiltered_noMiss_polym.vcf
```

3. En utilisant `vcftools`, nous allons ensuite créer des fichiers contenant les comptages de tous les allèles possibles, pour chaque site, sur l'ensemble des 15 individus.

```
vcftools --gzvcf 7-N00135_15ind_allsites_finalFiltered_noMiss_polym.vcf.gz --counts \
--out 7-N00135_15ind_allsites_finalFiltered_noMiss_polym
```

4. Refaites ce même calcul pour la séquence N00208. Inspectez les résultats avec la commande `head`. Que représentent les colonnes des fichiers obtenus ?
5. Nous allons ensuite traiter ces fichiers pour afficher la position des SNPs, les comptages des différents allèles et les comptages totaux. Pour ceci, nous allons utiliser un langage de programmation appelé `awk`, qui est très efficace pour le traitement de texte.

```
awk -v OFS='\t' '{{ split($5, all1, ":"); split($6, all2, ":"); print $2,
all1[2], all2[2], $4 }}' \
7-N00135_15ind_allsites_finalFiltered_noMiss_polym.frq.count > \
7-N00135_15ind_allsites_finalFiltered_noMiss_polym_sel.frq.count
```

Nous pouvons faire la même manipulation dans R. Ceci ne sera peut-être pas faisable en pratique sur des gros fichiers, mais sur cet exemple cela nous permet de mieux comprendre la manipulation du fichier.

```
counts <- read.table("7-N00135_15ind_allsites_finalFiltered_noMiss_polym.frq.count", skip=1, h=F)
head(counts)

## definition d'une nouvelle fonction pour traiter
get_counts <- function(x){
    return(strsplit(x, split=":")[[1]][2])
}

## application a tout le tableau avec sapply
counts$ComptageA1 <- sapply(counts$V5, get_counts)
counts$ComptageA2 <- sapply(counts$V6, get_counts)

## selection des colonnes
res <- counts[, c("V2", "ComptageA1", "ComptageA2", "V4")]

## ecriture du fichier de sortie
write.table(res, file="7-N00135_15ind_allsites_finalFiltered_noMiss_polym_sel.frq.count",
sep="\t", quote=F, row.names=F, col.names=F)
```

6. A partir de ce dernier fichier créé, récupérez la fréquence de l'allèle mineur et rajoutez une colonne au fichier de sortie, contenant uniquement la valeur "NA". Cette colonne correspond au taux de recombinaison des régions, qui est une information demandée par le programme (BalLeRMix) que nous allons utiliser à la fin du TP. Dans ce cas précis le taux de recombinaison n'est pas connu, d'où la valeur NA. BalLeRMix a besoin en entrée d'un fichier contenant 4 colonnes séparées par la tabulation : la position sur le chromosome, le taux de recombinaison (ici NA), le nombre de fois qu'on a vu l'allèle mineur, le nombre total de chromosomes (30 dans ce cas précis).

```
awk -v OFS='\t' '{{ if($3<$2) print $1, "NA", $3, $4; else print $1, "NA", \
$2, $4 }}' 7-N00135_15ind_allsites_finalFiltered_noMiss_polym_sel.frq.count \
> N00135_15ind_variants_sites_MAsorted.frq.count
```

Pour plus de lisibilité, nous pouvons faire le même traitement en R.

```
frq <- read.table("7-N00135_15ind_allsites_finalFiltered_noMiss_polym_sel.frq.count", h=F,
    stringsAsFactors=F)

head(frq)

## frequence de l'allele mineur
frq$maf <- apply(frq[, 2:3], 1, min)

## rajout info recombinaison
frq$rec <- rep(NA, nrow(frq))

res <- frq[,c("V1", "rec", "maf", "V4")]

write.table(res, "N00135_15ind_variants_sites_MAsorted.frq.count", sep="\t", quote=F, row.names=F,
    col.names=F)
```

7. Refaites tout ce traitement pour la séquence N00208, avec la procédure de votre choix pour la fin (awk ou R).

1.2 Description de la variabilité génétique

1. Nous allons d'abord examiner le spectre des fréquences alleliques (site frequency spectrum en anglais ou SFS). Pour ce faire, nous allons charger les données obtenues à l'étape précédente dans R et nous allons tracer l'histogramme des fréquences des allèles mineurs, sur l'ensemble des sites analysés.

```
count <- read.table("N00135_15ind_variants_sites_MAsorted.frq.count", h=F)

png(paste("SFS_plot_N00135.png", sep=""))

hist(count$V3, xlab="frequence de l'allele mineur", main="N00135", breaks=15)

dev.off()
```

2. Refaites cette analyse pour la séquence N00208.
3. Récupérez les images sur vos machines locales et comparez les résultats obtenus pour les deux séquences.
4. Nous allons ensuite calculer la densité en SNPs par fenêtre de 1 kilobase (kb). Pour ce faire, nous allons utiliser un programme "fait maison" en `awk`, (callable_sites.awk) qui permet d'identifier les positions informatives. En effet, la densité en SNPs doit se calculer par rapport aux sites informatifs, et non pas par rapport à toute la longueur de la séquence (qui peut comprendre des Ns, des régions répétées ou impossibles à aligner, etc).

```
mv ../Variant_Calling/callable_sites.awk .
```

```
vcftools --gzvcf 5-N00135_15ind_allsites_finalFiltered.vcf.gz \
--window-pi 1000 --out N00135_15ind_allsites_finalFiltered

vcftools --gzvcf 5-N00135_15ind_allsites_finalFiltered.vcf.gz \
--site-pi --out N00135_15ind_allsites_finalFiltered

awk -f callable_sites.awk N00135_15ind_allsites_finalFiltered.sites.pi \
>  N00135_15ind_allsites_finalFiltered.callable1000
```

5. Refaites ce traitement pour la séquence N00208.

1.3 Détection de sélection balancée avec BalLeRMix

1. Pour cette partie, nous allons utiliser l'outil BalLeRMix, qui se présente sous la forme d'un programme en Python, qui est fourni dans le dossier que vous avez téléchargé. Nous allons commencer par déplacer ce script dans le répertoire courant.

```
mv ../Variant_Calling/BalLeRMix_v2.3.py .
```

2. Nous allons ensuite faire tourner BaLeReMix sur les données traitées précédemment, pour la séquence N00135.

```
python3 BalLeRMix_v2.3.py -i N00135_15ind_variants_sites_MAsorted.frq.count \
--spect N00135_15ind_variants_sites_MAsorted.SFS --MAF --getSpect

python3 BalLeRMix_v2.3.py -i N00135_15ind_variants_sites_MAsorted.frq.count \
--spect N00135_15ind_variants_sites_MAsorted.SFS \
-o N00135_15ind_variants_sites_MAsorted.BM.out --MAF --nosub --physPos
```

3. Refaites la même analyse pour la séquence N00208.
4. Nous allons ensuite charger les résultats dans R, d'abord pour la séquence N00135.

```
vcf <- read.table("5-N00135_15ind_allsites_finalFiltered.ldepth.mean", h=T)

pi <- read.table("5-N00135_15ind_allsites_finalFiltered.windowed.pi", h=T)

BalSel <- read.table("N00135_15ind_variants_sites_MAsorted.BM.out", h=T)

callable_sites <- read.table("N00135_15ind_allsites_finalFiltered.callable1000", h=F)
```

5. Nous allons ensuite combiner toutes ces informations dans un même tableau (`data.frame`) dans R. Cela nous donnera toutes les informations nécessaires pour représenter graphiquement les données intéressantes (la densité en SNPs par fenêtre génomique, l'indice de diversité π, le score BalLeRMix etc.).

```
ii<-as.numeric(rownames(callable_sites))

window <- (ii-1)*1000+1

data.CS <- data.frame(window=window, callable_sites=callable_sites$V1)

my.pi <- merge(data.CS, pi, by.x = 1, by.y = 2)

my.pi$corDens <- my.pi$N_VARIANTS/my.pi$callable_sites

my.pi$corPi <- my.pi$PI/my.pi$callable_sites*100
```

6. Nous pouvons maintenant visualiser les résultats. Dans un premier temps, nous allons représenter le score BalLeRMix en fonction de la couverture.

```
png("BalLeRMix_Coverage_N00135.png")

par(mar = c(5, 4, 4, 4) + 0.3)

plot(vcf$POS, vcf$MEAN_DEPTH, cex=0.1, pch=1, xlab="", ylab="",
   col="black", ylim=c(0,200), axes=FALSE, bty="n")

axis(side=4, at = pretty(c(0,200)))

mtext("Coverage", side=4, line=3)

par(new=TRUE)

plot(BalSel$physPos, BalSel$LR, cex=0.4, xlab= "Position", ylab="Score BalLeRMix", col="turquoise",
    pch=16, ylim=c(0,500))

legend("topright", legend=c("BalLeRMix score", "Coverage"), col=c("turquoise","black"),
    pch=c(16,16))

dev.off()
```

7. Ensuite, nous allons représenter le score BalLeRMix en fonction de la densité en SNPs.

```
png("BalLeRMix_SNPdensity_N00135.png")

par(mar = c(5, 4, 4, 4) + 0.3)

plot((my.pi$window+my.pi$BIN_END)/2, my.pi$corDens, cex=0.1, pch= 1,
    xlab="", ylab="", col="black", ylim=c(0,1), axes=FALSE, bty="n")

axis(side=4, at = pretty(c(0,1)))

mtext("SNP density", side=4, line=3)

par(new=TRUE)

plot(BalSel$physPos, BalSel$LR, cex=0.4, xlab="Position physique", ylab="Score BalLeRMix", col="
     turquoise", pch=16, ylim=c(0,500))

legend("topright", legend=c("BalLeRMix score", "SNP density"),
     col=c("turquoise","black"), pch=c(16,16))

dev.off()
```

8. Rapatriez les images obtenues et visualisez-les. Que remarquez-vous ?
9. Refaites toute cette analyse pour la séquence N00208. Comment peut-on interpréter les différences obtenues pour les deux séquences ?