Code

AMMS Pratique : Résistance aux médicaments

Andres Aranda-Diaz, Lucy Okell, Bob Verity

2022-08-04

Dépendances pour Pratique

Veuillez copier et coller le morceau de code ci-dessous dans son intégralité sur votre console pour télécharger les bibliothèques de packages R nécessaires à cette pratique. Si vous rencontrez des difficultés pour installer l'un des packages R, veuillez demander à un instructeur un lecteur flash préchargé.

Hide

```
if (!("tidyverse" %in% installed.packages())) {
  install.packages("tidyverse", dependencies = TRUE)
}
```

Chargez maintenant toutes ces bibliothèques dans cette session en utilisant le morceau de code ci-dessous. Veuillez le copier-coller dans son intégralité.

Hide

```
library(tidyverse)
```

Introduction à la résistance aux médicaments

La résistance aux médicaments est l'un des principaux cas d'utilisation de la surveillance moléculaire du paludisme. En quantifiant la fréquence des mutations connues pour conférer une résistance aux médicaments couramment utilisés, nous obtenons un système d'alerte précoce efficace qui peut être suivi d'études d'efficacité thérapeutique (TES) pour établir le risque d'échec clinique. Les données sur la résistance aux médicaments peuvent être assez simples, consistant en un numérateur et un dénominateur qui peuvent être utilisés pour estimer la prévalence, mais il existe également certaines complexités avec des combinaisons de mutations et d'haplotypes résistants aux médicaments que nous devons également explorer.

Aperçu des données

Dans cette pratique, nous utiliserons un ensemble de données du monde réel composé d'un grand ensemble de données de sonde d'inversion moléculaire (MIP) provenant de la RDC et des pays environnants, précédemment analysé par [Verity et al. 2020] (https://pubmed.ncbi.nlm.nih.gov/32355199/ (https://pubmed.ncbi.nlm.nih.gov/32355199/)). Nous utiliserons une version simplifiée de cet ensemble de données en nous concentrant sur seulement trois mutations du gène *dhps*.

Objectifs pratiques

À la fin de cet exercice pratique, vous devriez être en mesure de :

- Importer des données dans RStudio
- Décrire un ensemble de données génétiques
- Appliquer des fonctions tidyverse pour résumer les données
- Calculer la prévalence à partir de données basées sur des échantillons
- Visualiser les données de prévalence
- Construire un intervalle de confiance à 95%
- Visualiser les données d'haplotype

Chargement et résumé des données génétiques

Aperçu des données

Les données génétiques peuvent être agrégées ou par échantillon. Les informations contenues dans un ensemble de données simple par échantillon doivent inclure un identifiant d'échantillon, des informations sur le locus (position ou un autre identifiant, séquence de référence et alternative) et l'abondance de séquences de référence et alternatives dans ce locus. Les métadonnées peuvent également être incluses. Pour les échantillons, nous souhaiterons peut-être savoir quand et où ils ont été collectés, ainsi que toute autre information pertinente. Pour les loci, nous pouvons vouloir connaître, par exemple, les séquences d'acides aminés pour lesquelles codent les codons.

Dans ce TP, nous utiliserons un ensemble de données collectées en République Démocratique du Congo en 20?. Nous avons simplifié l'ensemble de données d'origine à des fins pédagogiques. Nous examinerons les mutations liées à la résistance dans le gène dhps.

Se familiariser avec les données

Chargeons l'ensemble de données :

Hide

```
dr.data <- readRDS("data/DRC_DR2.rds")
```

Pouvez-vous voir dr.data dans votre environnement ?

Pour vous familiariser avec l'ensemble de données, répondons aux questions suivantes :

Q1. Combien de lignes et de colonnes y a-t-il dans dr.data ? Quels sont les noms de colonnes ?
Click For Answer

A1.

Hide

```
ncol(dr.data)
```

```
## [1] 13
```

Hide

```
colnames(dr.data)
```

```
##  [1] "SAMPLE_ID" "REGION"    "CHROM"     "POS"       "REF"       "ALT"
##  [7] "GENE_NAME" "CODON_NUM" "CODON"     "CODON_POS" "REF_AA"    "ALT_AA"
## [13] "REF_WSAF"
```

Hide

```
nrow(dr.data)
```

```
## [1] 1617
```

Nous pouvons voir que nous avons 13 colonnes et 1617 lignes. Les colonnes contiennent un identifiant d'échantillon (SAMPLE_ID) et des métadonnées d'échantillon (REGION), ainsi que des informations sur le lieu (le reste des colonnes). La dernière colonne est appelée REF_WSAF et décrit la proportion de séquences de référence dans un locus et un échantillon donnés.

Jetons maintenant un coup d'œil à dr.data. Parce que dr.data a 1617 lignes, il est trop long à afficher, n'affichons que les 10 premières lignes.

Hide

```
dr.data[1:10,]
```

```
##    SAMPLE_ID REGION CHROM    POS REF ALT GENE_NAME CODON_NUM CODON CODON_POS
## 1  1025N6S5R   EAST     8 549685   G   C      dhps        437   GGT         1
## 2  1025N6S5R   EAST     8 549993   A   G      dhps        540   AAA         0
## 3  1025N6S5R   EAST     8 550117   C   G      dhps        581   GCG         1
## 4  1032V2A9H   EAST     8 549685   G   C      dhps        437   GGT         1
## 5  1032V2A9H   EAST     8 549993   A   G      dhps        540   AAA         0
## 6  1032V2A9H   EAST     8 550117   C   G      dhps        581   GCG         1
## 7  1145J7T1L   WEST     8 549685   G   C      dhps        437   GGT         1
## 8  1145J7T1L   WEST     8 549993   A   G      dhps        540   AAA         0
## 9  1145J7T1L   WEST     8 550117   C   G      dhps        581   GCG         1
## 10 1152C2M8Y   EAST     8 549685   G   C      dhps        437   GGT         1
##    REF_AA ALT_AA  REF_WSAF
## 1       A      G 0.0000000
## 2       K      E 1.0000000
## 3       A      G 1.0000000
## 4       A      G 0.6666667
## 5       K      E        NA
## 6       A      G        NA
## 7       A      G 0.2058824
## 8       K      E 1.0000000
## 9       A      G 1.0000000
## 10      A      G 1.0000000
```

Comme vous pouvez le constater, chaque échantillon comporte plusieurs lignes. Cela signifie probablement que nos données sont dans un format long, où les informations pour plusieurs lieux d'un échantillon sont contenues dans différentes lignes, au lieu de colonnes. En outre, certaines lignes affichent des valeurs NA dans REF_WSAF. Cela signifie que dans cet échantillon, il n'y a aucune information dans ce lieu.

Q2. Pourquoi voudrions-nous stocker les données dans un format long ?

Click For Answer

A2. Le format long est utile lors du stockage de valeurs irrégulières ou manquantes, et est généralement considéré comme plus efficace. Un avantage majeur pour nos besoins est qu'il nous permet d'utiliser les fonctions `ggplot2` , qui attendent des données au format long en entrée.

Maintenant, obtenons plus d'informations sur le contenu de dr.data en répondant aux questions suivantes :

Combien de codons regardons-nous ? Combien y a-t-il d'échantillons uniques ? Combien y a-t-il d'échantillons par région ?

Pour résumer les données, les outils de la bibliothèque tidyverse sont très utiles. Certains de ces outils incluent les fonctions `distinct` , `unique` , `select` , `group_by` et `summarise` . Pour obtenir plus d'informations sur une fonction, accédez à la documentation en utilisant ?function. Copiez `?distinct` dans votre console.

Utilisons les fonctions `unique` et `length` pour identifier le nombre d'échantillons dans dr.data

Hide

```
# get unique IDs
samples <- unique(dr.data$SAMPLE_ID)

# calculate number of such IDs
length(samples)
```

```
## [1] 539
```

Utilisons la fonction `distinct` pour identifier combien et quels codons sont contenus dans dr.data.

Hide

```
# First, let's select relevant columns for this question: GENE_NAME, CODON_NUM
dr.data.loci <- dr.data[,c("GENE_NAME","CODON_NUM")]

# Next, lets select distinct rows
dr.data.loci <- distinct(dr.data.loci)

# let's display the result
dr.data.loci
```

```
##   GENE_NAME CODON_NUM
## 1      dhps       437
## 2      dhps       540
## 3      dhps       581
```

Nous pouvons voir qu'il y a 3 codons uniques pour le gène *dhps* dans l'ensemble de données.

Nous avons effectué deux actions sur dr.data : (1) nous avons sélectionné les colonnes pertinentes et (2) nous avons sélectionné uniquement des lignes distinctes du bloc de données. Nous pouvons également utiliser des pipes pour effectuer des actions séquentielles sur un objet. Le code suivant effectue exactement les mêmes opérations :

Hide

```
dr.data.loci <- dr.data %>%
  select(GENE_NAME,CODON_NUM) %>%
  distinct()

dr.data.loci
```

```
##   GENE_NAME CODON_NUM
## 1      dhps       437
## 2      dhps       540
## 3      dhps       581
```

Utilisons maintenant les canaux et les fonctions `select`, `distinct`, `group_by` et `summarise` pour obtenir un nombre d'échantillons par REGION

Hide

```
dr.data.per.region <- dr.data %>%
  select(SAMPLE_ID,REGION) %>%
  distinct() %>%
  group_by(REGION) %>%
  summarise(n = n())

dr.data.per.region
```

```
## # A tibble: 2 × 2
##   REGION     n
##   <chr>  <int>
## 1 EAST     296
## 2 WEST     243
```

Estimation de la prévalence

Dans les prochaines sections, vous comparerez les résultats avec d'autres membres de votre groupe. Chacun de vous se concentrera sur un codon : Participants 1 : 437 Participants 2 : 540 Participants 3 : 581

Pour la pharmacorésistance, la prévalence d'un mutant est définie comme la proportion d'observations dans une population qui contiennent une mutation. Étant donné que les infections paludéennes peuvent être polyclonales, un individu peut être infecté par un mélange de souches possédant des allèles de type sauvage ou mutants. Ainsi, nous devons définir comment nous comptabiliserons les observations.

Une façon d'y parvenir consiste à compter la prévalence des infections de type sauvage, mutantes et mixtes. Alternativement, nous pouvons compter chaque parasite comme une observation (ce qui signifie qu'une infection polyclonale comptera comme plusieurs observations).

Nous utiliserons cette dernière définition. Calculons d'abord la prévalence globale des mutants résistants aux médicaments dans l'ensemble du pays.

Comme nous l'avons expliqué ci-dessus, chaque échantillon a une ligne pour chaque locus, et la variable "REF_WSAF" décrit la proportion d'observations de référence (type sauvage) dans un échantillon. `REF_WSAF = 1` signifie que l'échantillon est exclusivement de type sauvage et `REF_WSAF = 0` signifie qu'il est exclusivement mutant. Toute valeur entre 0 et 1 indique qu'il y a un mélange des deux allèles dans l'échantillon. Ainsi, nous pouvons compter le nombre d'échantillons contenant une observation de type sauvage comme ceux avec 'REF_WSAF > 0', et ceux contenant une observation mutante avec `REF_WSAF < 1`.

Tout d'abord, filtrons les données pour contenir les lignes correspondant au codon qui vous a été attribué et créons 2 variables indiquant si les allèles de type sauvage et mutant sont présents dans l'échantillon. Dans le code suivant, veuillez remplacer l'assigned_codon par votre codon correspondant

Hide

```
assigned_codon <- 540
dr.data.filtered <- dr.data %>%
  filter(CODON_NUM == assigned_codon) %>%
  mutate(REF_ALLELE = REF_WSAF > 0) %>%
  mutate(ALT_ALLELE = REF_WSAF < 1)
```

Calculons le nombre total d'observations de type sauvage

Hide

```
sum(dr.data.filtered$REF_ALLELE)
```

```
## [1] NA
```

La valeur est 'NA' car, comme nous l'avons expliqué ci-dessus, certains des échantillons n'ont aucune information pour ce lieu, et la plupart des opérations renverront un 'NA' si au moins 1 valeur est 'NA'. Nous devons ensuite spécifier que nous voulons supprimer les NA :

Hide

```
sum(dr.data.filtered$REF_ALLELE, na.rm = TRUE)
```

```
## [1] 335
```

De même, on peut calculer le nombre d'observations mutantes :

Hide

```
sum(dr.data.filtered$ALT_ALLELE, na.rm = TRUE)
```

```
## [1] 185
```

Q3. Quelle est la prévalence de mutants dans votre codon ? Comment se compare-t-il aux autres marqueurs de votre groupe ?

Click For Answer

A3.

Hide

```
total_ALT <- sum(dr.data.filtered$ALT_ALLELE, na.rm = TRUE)
total_REF <- sum(dr.data.filtered$REF_ALLELE, na.rm = TRUE)

PREVALENCE <- total_ALT/(total_ALT + total_REF)
PREVALENCE
```

```
## [1] 0.3557692
```

Nous allons maintenant effectuer une analyse régionale.

Q4. Quelle est la prévalence de mutants dans votre codon dans les deux régions ? Comment se compare-t-il aux autres marqueurs de votre groupe ?

Click For Answer

A4.

Hide

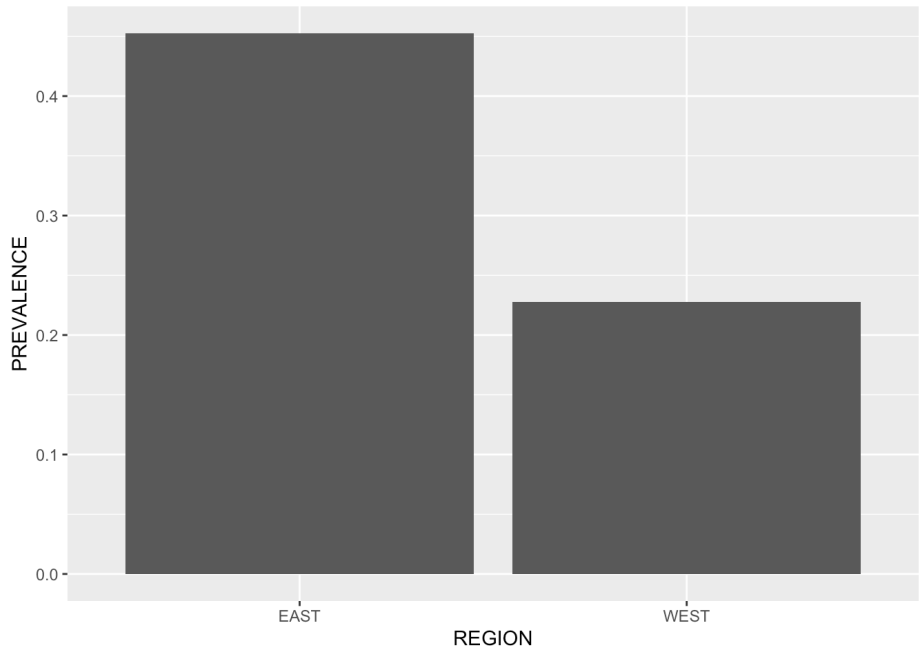
```
dr.prevalence.region <- dr.data.filtered %>%
  group_by(REGION) %>%
  summarise(total_ALT = sum(ALT_ALLELE, na.rm = TRUE),
            total_REF = sum(REF_ALLELE, na.rm = TRUE),
            PREVALENCE = total_ALT / (total_ALT + total_REF))
dr.prevalence.region
```

```
## # A tibble: 2 × 4
##   REGION total_ALT total_REF PREVALENCE
##   <chr>      <int>     <int>      <dbl>
## 1 EAST         134       162      0.453
## 2 WEST          51       173      0.228
```

Maintenant, faisons également un graphique à barres de vos données :

Hide

```
ggplot(data = dr.prevalence.region,
       aes(x = REGION, y = PREVALENCE))+
  geom_bar(stat="identity",position="dodge")
```



Q5. Où la prévalence est-elle la plus élevée, à l'est ou à l'ouest ?

Click For Answer

A5. Cela peut varier en fonction de votre codon, mais d'après le graphique à barres ci-dessus (pour le codon 540), il semble que la prévalence soit la plus élevée à l'est.

Calculer les intervalles de confiance

Chaque estimation de la prévalence comporte une certaine incertitude. Ici, vous allez calculer l'intervalle de confiance à l'aide de la formule d'intervalle de Wald.

La limite inférieure est calculée comme suit : $p - z\sqrt{\frac{p(1-p)}{n}}$

La borne supérieure est calculée comme suit : $p + z\sqrt{\frac{p(1-p)}{n}}$

Ainsi, nous devons connaître la prévalence (p), la taille de l'échantillon (n) et le niveau de confiance que nous voulons. Pour ce TP, nous calculerons des niveaux de confiance à 95 %, ce qui signifie que $z = 1,96 $.

Calculons d'abord l'intervalle de confiance pour le pays dans son ensemble :

Hide

```
total_ALT <- sum(dr.data.filtered$ALT_ALLELE, na.rm = TRUE)
total_REF <- sum(dr.data.filtered$REF_ALLELE, na.rm = TRUE)

N <- total_ALT + total_REF
PREVALENCE <- total_ALT / N

CI.LOWER <- PREVALENCE - 1.96 * sqrt(PREVALENCE * (1 - PREVALENCE) / N)
CI.UPPER <- PREVALENCE + 1.96 * sqrt(PREVALENCE * (1 - PREVALENCE) / N)

c(PREVALENCE, CI.LOWER, CI.UPPER)
```

```
## [1] 0.3557692 0.3146202 0.3969182
```

Q6. Est-ce que votre réponse à la question "Quel codon a une prévalence plus élevée ?" changer maintenant que vous avez vu des intervalles confiants ?

Click For Answer

A6. Nous constatons que les CI pour le codon 540 et le codon 437 se chevauchent. Cela signifie qu'il est difficile de dire avec certitude lequel d'entre eux a une prévalence plus élevée. Le codon 581 est inférieur et ne se chevauche pas, nous pouvons donc être sûrs qu'il s'agit bien de la prévalence la plus faible.

Calculons maintenant les intervalles de confiance pour la prévalence dans chaque région, ce qui sera plus facile avec les outils tidyverse :

Hide

```
dr.prevalence.region <- dr.data.filtered %>%
  group_by(REGION) %>%
  summarise(total_ALT = sum(ALT_ALLELE, na.rm = TRUE),
            total_REF = sum(REF_ALLELE, na.rm = TRUE),
            N = total_ALT + total_REF,
            PREVALENCE = total_ALT / N) %>%
  mutate(ci.lower = PREVALENCE - 1.96 * sqrt(PREVALENCE * (1 - PREVALENCE) / N),
         ci.upper = PREVALENCE + 1.96 * sqrt(PREVALENCE * (1 - PREVALENCE) / N))

dr.prevalence.region
```

```
## # A tibble: 2 × 7
##   REGION total_ALT total_REF     N PREVALENCE ci.lower ci.upper
##   <chr>      <int>     <int> <int>      <dbl>    <dbl>    <dbl>
## 1 EAST         134       162   296      0.453    0.396    0.509
## 2 WEST          51       173   224      0.228    0.173    0.283
```

Q7. Est-ce que votre réponse à la question « Dans quelle région y a-t-il une prévalence plus élevée ? » changer maintenant que vous avez vu les intervalles de confiance ? Qu'en est-il des autres codons de votre groupe ?

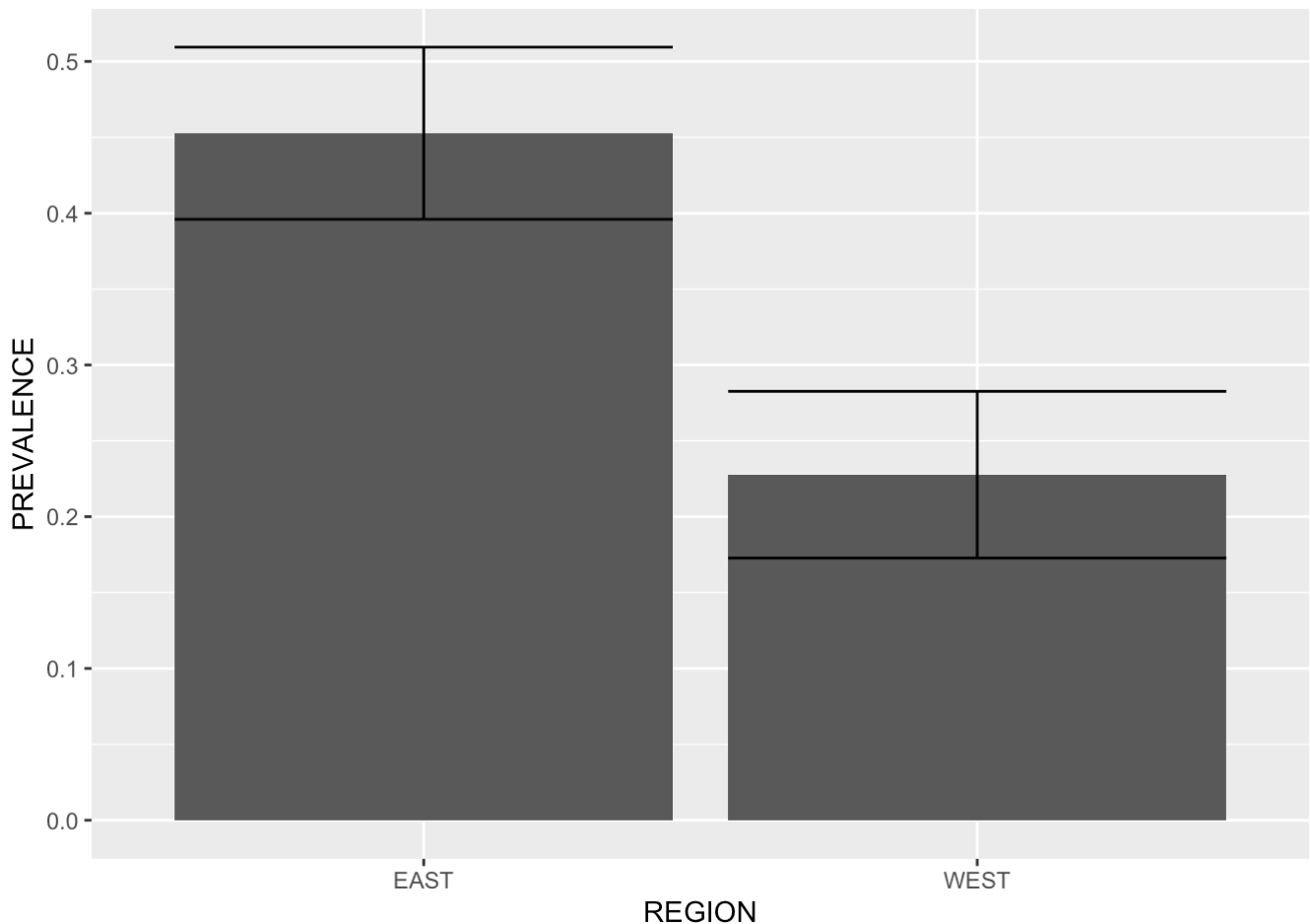
Click For Answer

A7. Cela peut varier en fonction de votre codon, mais d'après le graphique à barres ci-dessus (pour le codon 540), il semble que les intervalles de confiance ne se chevauchent pas. Cela signifie qu'il est probable que la prévalence soit vraiment plus élevée dans l'Est, et ce n'est pas seulement un effet d'échantillonnage.

Ajoutons maintenant ces nouvelles informations à notre visualisation :

Hide

```
ggplot(data = dr.prevalence.region,
       aes(x = REGION, y = PREVALENCE, ymin = ci.lower, ymax = ci.upper)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar()
```



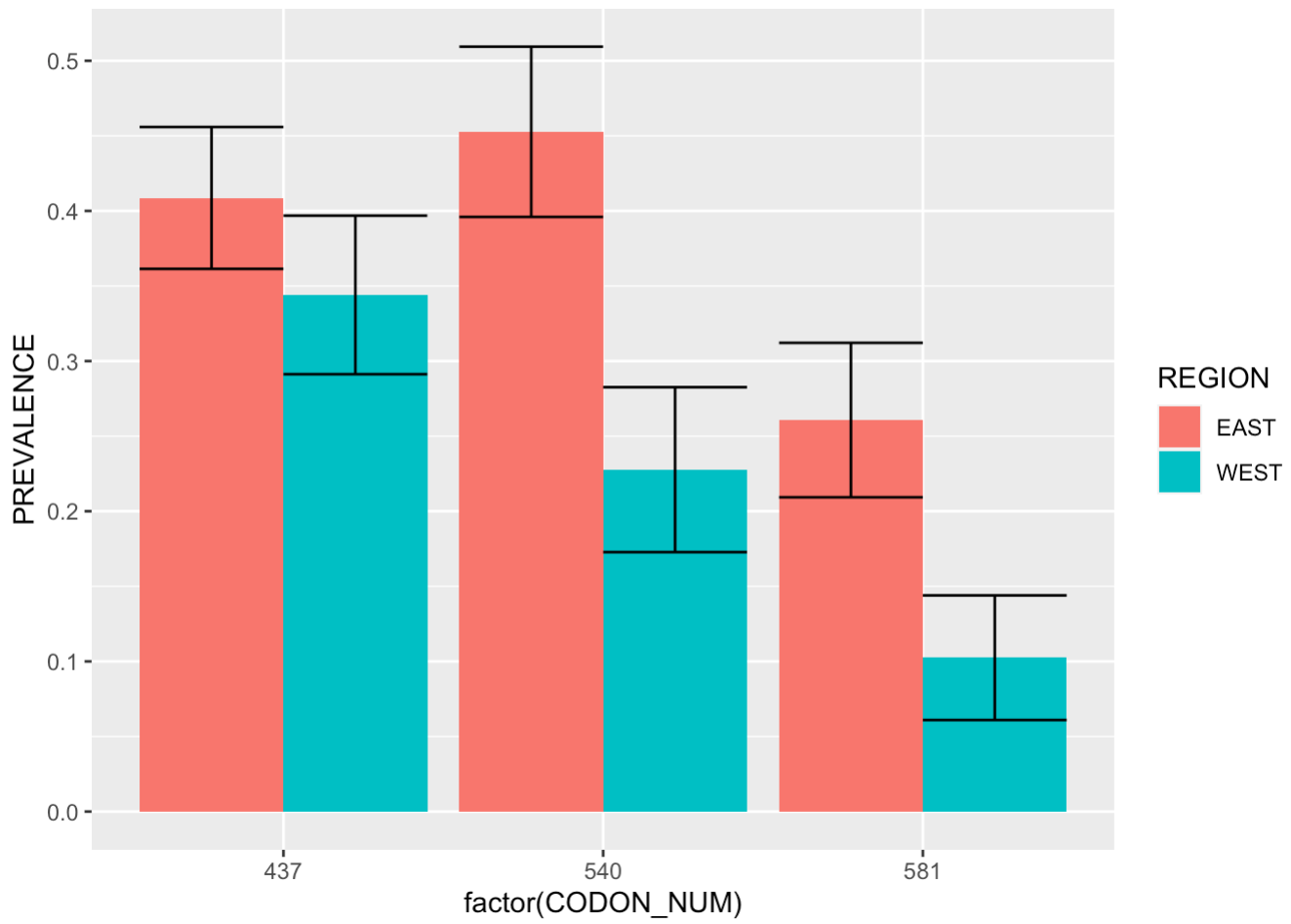
Intégration des données de plusieurs codons

Maintenant que chacun d'entre vous a travaillé sur un codon indépendamment, il est temps de les rassembler tous en une seule figure :

Hide

```
dr.data.grouped.wald <- dr.data %>%
  group_by(CODON_NUM, REGION) %>%
  summarise(total_ALT = sum(REF_WSAF < 1, na.rm = TRUE),
            total_REF = sum(REF_WSAF > 0, na.rm = TRUE),
            N = total_ALT + total_REF,
            PREVALENCE = total_ALT / N) %>%
  mutate(ci.lower = PREVALENCE - 1.96 * sqrt(PREVALENCE * (1 - PREVALENCE) / N),
         ci.upper = PREVALENCE + 1.96 * sqrt(PREVALENCE * (1 - PREVALENCE) / N))

ggplot(data = dr.data.grouped.wald,
       aes (fill = REGION,
            y = PREVALENCE,
            x = factor(CODON_NUM),
            ymin = ci.lower,
            ymax = ci.upper)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(position = "dodge")
```



Visualiser les haplotypes

Comme nous en avons discuté dans la conférence, pour certains médicaments, la combinaison d'allèles (haplotypes) est l'information la plus pertinente. Ensuite, nous utiliserons un package, `UpSetR` , pour visualiser les combinaisons de mutations. La documentation sur le package peut être trouvée ici (https://cran.r-project.org/web/packages/UpSetR/UpSetR.pdf).

Les données polyclonales présentent un défi pour la construction d'haplotypes. Par exemple, si un échantillon contient 2 allèles à chacun des 2 loci, comment pouvons-nous savoir quel allèle va avec lequel dans chacun des parasites de cette infection ? Ce problème est appelé **phasage**, et nous travaillons souvent avec des données **non phasées**.

Pour simplifier notre ensemble de données, nous supprimerons les échantillons avec "NAs" et ne considérerons que l'allèle le plus abondant dans chaque échantillon :

Hide

```
# identify NAs
dr.data.NA <- dr.data %>%
  group_by(SAMPLE_ID) %>%
  summarise(na = !any(is.na(REF_WSAF)))

# simplify by rounding within-sample allele frequencies to 0 or 1
dr.data.simplified <- dr.data %>%
  mutate(REF_WSAF_ROUND = round(REF_WSAF)) %>%
  filter(SAMPLE_ID %in% (dr.data.NA %>% filter(na))$SAMPLE_ID)

# count the number of each haplotype
dr.data.simplified %>%
  select(SAMPLE_ID, CODON_NUM, REF_WSAF_ROUND) %>%
  pivot_wider(names_from = CODON_NUM , values_from = REF_WSAF_ROUND) %>%
  mutate(haplotype_437_540_581 = paste(`437`,`540`,`581`)) %>%
  group_by(haplotype_437_540_581) %>%
  summarize(n = n())
```

```
## # A tibble: 6 × 2
##   haplotype_437_540_581     n
##   <chr>                 <int>
## 1 0 0 0                     1
## 2 0 0 1                     1
## 3 0 1 1                   126
## 4 1 0 0                    44
## 5 1 0 1                    76
## 6 1 1 1                   162
```

Nous pouvons voir que certaines combinaisons d’haplotypes sont plus courantes que d’autres. Par exemple, le triple mutant est très courant, tout comme le double mutant 540+581.