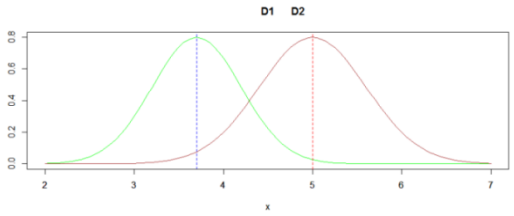
Cours VII. Tests de randomisation - Tests de contingence

P. Coquillard 2015

TESTS DE RANDOMISATION

Dans une majorité de cas en biologie on considèrera certaines hypothèses comme des alternatives à l'hypothèse nulle. En réalité, l'hypothèse étudiée évoque une structure ayant tendance à apparaitre dans les données disponibles, alors que l'hypothèse nulle nous dit que si cette structure est présente c'est seulement *le fruit du pur hasard* de l'échantillonnage.

Les tests de randomisation sont particulièrement utiles lorsque l'on a à comparer des échantillons ne vérifiant pas la normalité de leurs distributions et/ou qu'ils sont petits.



Comparaison de 2 distributions. H0: D1 = D2 ; H1: D1 ≠ D2

Les tests de randomisation sont une manière de décider si l'hypothèse nulle est acceptable en de telles situations. Une statistique S est choisie pour évaluer dans quelle mesure les données présentent la structure en question. L'estimation s de **S** obtenue à partir des données est alors comparée avec la distribution de **S** obtenue en *réordonnant au hasard* (permutations) les données. L'idée est simplement que si l'hypothèse nulle est vraie, alors toutes les combinaisons possibles des données sont équiprobables. Les données observées sont alors seulement l'une des réalisations parmi toutes celles également possibles et *s* est une valeur typique de la distribution aléatoire de **S**. Si tel n'est pas le cas (*s* est significativement différente), l'hypothèse H_0 est rejetée et H_1 considérée comme plus vraisemblable.

Le niveau de signification de *s* est simplement la proportion (%) de valeurs trouvées dans la distribution obtenue par permutation qui sont aussi extrêmes ou plus extrêmes que cette valeur. Avec R, les fonctions utiles pour réaliser un test de permutation sont :

1) `D <- sample (C, length(C), replace = FALSE)`, où C est la concaténation des deux échantillons, (`C <-c(A,B)`), et « replace » est mis à FALSE, ce qui impose que *tous les éléments sont tirés sans remise*. Le nombre de tirages différents est `factorial(length(C))`. Soit un échantillon de taille 5, le nombre de tirage possible est alors de 5 ! = 5x4x3x2=120.

Si A contient n_1 données et B n_2 données, on constitue deux nouveaux échantillons de tailles respectives n_1 et n_2 à partir de D :

2) `A.random = D[1 : length(A)]`
3) `B.random = D[(length(A) + 1) : length(C)]`
4) On calcule ensuite la différence des moyennes et on les stocke dans un tableau :

```
diff.random[i] = mean(A.random) - mean(B.random)
```

Les opérations 1 à 4 sont répétées 1000 fois au moins. Il reste maintenant à comparer ces différences avec celle mesurées sur les données initiales :

`p = sum(abs(diff.random)>= abs(diff.observe))/1000`, Ce qui est la p-value.

La méthode des permutations est parfois utilisée en phylogénie moléculaire (Archie (1989 ; Faith et Cranston 1991) en l'appliquant sur les colonnes des séquences alignées.

Le but est de répondre à la question s'il existe ou non (H0) un lien phylogénétique entre les séquences étudiées.

Algorithme du bootstrapping

Le problème est de connaitre les paramètres d'une statistique : espérance, moyenne, écart-type, voire des intervalles de confiance à partir d'un petit échantillon sans information complémentaire autre que celles disponibles à partir de l'échantillon. On utilise une technique de ré-échantillonnage.

Soit un échantillon x constitué de n observations (X1, X2,…Xn) et θ un paramètre (médiane, moyenne…) à estimer. *Toutefois la distribution F des observations est inconnue.* On a donc à estimer $\theta = T(F)$, où T est une fonctionnelle.

Pour la moyenne :$T(F) = \int x dF(x)$ et la variance : $T(F) = \int (x-\mu)^2 dF(x)$

On ne fait aucune hypothèse sur F qui est inconnue. Pour cette raison on effectuera un bootstrap *non paramétrique* qui consiste simplement en un ré-échantillonnage *avec remise* dans l'échantillon initial. Soit un échantillon de 5 éléments, le nombre de tirages possibles avec remise est alors de $5^5 = 3125$.
Par exemple, si l'on dispose de *n* valeurs initiales, on tirera *n* valeurs parmi ces *n* valeurs avec remise après chacun des tirages. En conséquence, l'une des *n* valeurs de l'échantillon initial peut être tirée plusieurs fois et certaines valeurs de celui-ci être absentes de ce nouvel échantillon. On dispose donc maintenant de deux échantillons : l'initial (issu d'une expérimentation) et celui du bootstrap. En réalité, 2 échantillons ne sont pas suffisants. On va réitérer cette opération un grand nombre de fois pour être assuré de la convergence des estimations que l'on va faire à partir de l'ensemble des échantillons ainsi constitués. Soit *B* ce nombre (grand : en général 1000 est conseillé).
Algorithme pour l'estimation de la variance de la loi (sa précision) :

- Boucle : pour *b* allant de 1 à *B* :
  - Tirer un échantillon bootstrap: X1, X2,…,Xn selon *F* et de taille *n.*
  - Calculer la moyenne empirique à partir de l'échantillon bootstrap :

$$\hat{\theta} = \frac{1}{n}(X1 + X2 + \cdots + Xn)$$

- La variance de l'estimateur de l'espérance est approchée par la variance empirique de la population bootstrap des $B\hat{\theta}$ estimés, soit :

$$s_B^2 = \frac{1}{B}\sum_{b=1}^{B}\left[\hat{\theta}_b - \bar{\hat{\theta}}\right]^2 \text{ avec } \bar{\hat{\theta}} = \frac{1}{B}\sum_{b=1}^{B}\widehat{\theta_b}$$

Pour le calcul d'un intervalle de confiance autour de la moyenne des *B* échantillons, il suffira de calculer les pourcentiles à 2.5% et à 97.5% qui nous donnerons les bornes inférieures et supérieures de l'intervalle autour de la moyenne. Avec R l'échantillon bootstrap est obtenu ainsi :

```
library(boot)
B <- boot(data, fonction, R = 999, stype = "f")
```

où « stype » indique la nature du second argument de la fonction (i = indice, f = frequence, w = weight) qui calcule le paramètre cherché (moyenne, écart-type, médiane, etc…), son premier argument étant les données. « R » indique le nombre de réplicats du tirage.

Applications :

1). Voir le script R (UE10) : Bootstrap (R)

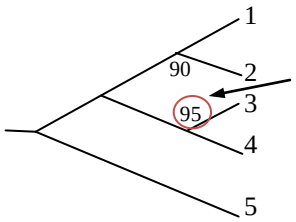
2). Test de robustesse de reconstruction des arbres phylogénétiques (Felsenstein, 1985). *Elle est indépendante de la méthode de construction* (distance génétique (UPGMA ou Neighbor-Joining), parcimonie, vraisemblance (ML)…)

Interprétation des valeurs b de bootstrap = probabilité que la longueur de la branche soit > 0. La branche est réputée significative si b > 95% (mais cela dépend des auteurs). *Cette valeur ne dit rien à propos de la longueur de la branche qui dépend de la méthode de construction de l'arbre*.

Tirage aléatoire avec remise des n colonnes initiales des séquences alignées.

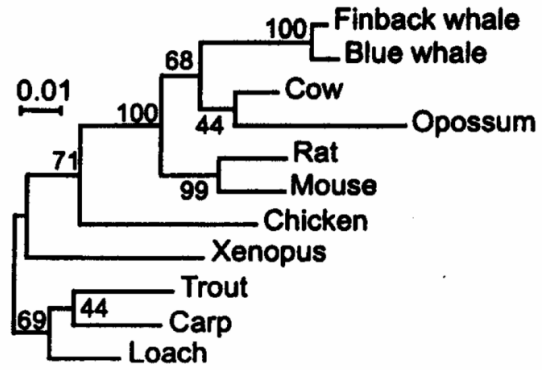
Données initiales		Après ré échantillonnage	
Taxa	**12345678**…		**64673752**…
1	CGAGTACT…	1	AGATACTG…
2	GTAGTACT…	2	AGATACTT…
3	ACAATACC…	3	AAACACTC…
4	ACAACACT…	4	AAATACCC…
5	GCGGCATT…	5	AGATGTCC…

Un total de 100 jeux (au moins),
⇨ 100 arbres construits dont :
90 présentent le clade (1,2)
95 présentent le clade (3,4)



Proportion d'arbres présentant ce nœud

= niveau de confiance



ex. arbre consensus sur 500 bootstraps avec méthode ML sur données de séquences de la cytochrome oxidase (gène mitochondrial)

(Nei & Kumar, 2000)

Trout = truite, *Loach* = loche (poissons marins de divers genres), *Xenopus* = Xénope = « grenouilles » tropicales, *Finback whale* = Rorqual commun.

TABLES DE CONTINGENCE

a. **Le tableau de contingence** est un tableau à double entrée où l'on note dans chaque cellule le nombre d'individus ayant à la fois le caractère 1 (ligne) et le caractère 2 (colonne). Chaque caractère peut avoir plusieurs modalités. Soit 2 caractères L et C, variant sur $p = q = 3$ modalités, le tableau se présentera ainsi :

Caractères : **L/C**	**1**	**2**	**3**	(*p colonnes*)
1	n_{11}	n_{12}	n_{13}	$\boldsymbol{n_{i.} = \sum_{j=1}^{p} n_{ij}}$
2	n_{21}	n_{22}	n_{23}	$\boldsymbol{n_{i.} = \sum_{j=1}^{p} n_{ij}}$
3	n_{31}	n_{32}	n_{33}	$\boldsymbol{n_{i.} = \sum_{j=1}^{p} n_{ij}}$
(*q lignes*)	$\boldsymbol{n_{.j} = \sum_{i=1}^{q} n_{ij}}$	$\boldsymbol{n_{.j} = \sum_{i=1}^{q} n_{ij}}$	$\boldsymbol{n_{.j} = \sum_{i=1}^{q} n_{ij}}$	*Effectifs marginaux*

L'effectif total du tableau est alors: $\boldsymbol{n = \sum_{i=1}^{q}\sum_{j=1}^{p} n_{ij}}$

De plus, chaque élément du tableau peut être vu comme une fréquence puisque :

$$\boldsymbol{f_{ij} = \frac{n_{ij}}{n}}, \text{ avec bien entendu : } \boldsymbol{\sum_{i=1}^{q}\sum_{j=1}^{p} f_{ij} = 1}$$

Les rapports des effectifs marginaux au total sont intéressants. Ainsi $\left(\frac{n_{i.}}{n}\right)$ est la probabilité qu'un individu ait le caractère *i* et $\frac{n_j}{n}$ qu'un individu possède le caractère *j*.

De plus sont des probabilités conditionnelles : $\left(\frac{n_{ij}}{n_{i.}}\right)$ et $\left(\frac{n_{ij}}{n_{.j}}\right)$, c'est-à-dire, dans le premier cas, la probabilité qu'un individu ait le caractère *j* sachant qu'il possède le caractère *i*, notée :

$$\frac{\boldsymbol{n_{ij}}}{\boldsymbol{n_{i.}}} = \wp(\boldsymbol{C_j}|\boldsymbol{L_i})$$

c'est-à-dire, la probabilité qu'un individu ait le caractère « Colonne *j* » sachant qu'il a le caractère « Ligne *i* ». On a identiquement : $\wp(\boldsymbol{L_j}|\boldsymbol{C_i})$.

Avec une petite manipulation algébrique on retrouve la formule de Bayes:

$$\wp(\boldsymbol{L_j}|\boldsymbol{C_i}) = \frac{\frac{\boldsymbol{n_{ij}}}{\boldsymbol{n_{i.}}} \times \frac{\boldsymbol{n_{i.}}}{\boldsymbol{n}}}{\frac{\boldsymbol{n_{.j}}}{\boldsymbol{n}}} = \frac{\wp(\boldsymbol{C_j}|\boldsymbol{L_i})\wp(\boldsymbol{L_i})}{\wp(\boldsymbol{C_j})}$$

⇨ Pour tester l'existence d'une relation entre les distributions L et C ou si au contraire la répartition des valeurs se fait indépendamment de celles-ci, on définira un modèle nul puis on cherchera ensuite si la répartition observée diffère significativement de ce modèle nul.

b. Etablissement du modèle nul. On construit donc une nouvelle table en supposant que chaque cellule $\boldsymbol{\hat{n}_{ij}}$est calculable à partir des effectifs marginaux, c'est-à-dire :

$$\boldsymbol{\hat{n}_{ij} = \frac{n_{i.}n_{.j}}{n}}$$

c. Test du χ^2 de Pearson (toutes tailles de tables) R : `chisq.test(data)`

Le test consiste à calculer la distance entre les données et les valeurs calculées précédemment :

$$\boldsymbol{\sum_{i=1}^{q}\sum_{j=1}^{q} \frac{(\hat{n}_{ij} - n_{ij})^2}{\hat{n}_{ij}}}$$

Par définition, cette somme de carrés de variables centrées réduites suit une distribution du χ^2. On confronte donc cette valeur à celle de la loi du χ^2 pour (*p*-1)(*q*-1) degrés de libertés.

Il existe toutefois des limites d'application du test du χ^2. On donne souvent une limite concernant les effectifs théoriques : $(\widehat{n_{ij}} > 5) \vee (\widehat{n_{ij}} = 0), \forall i, \forall j$. On a alors recourt au test suivant.

d. Test G (toutes tailles de tables) `G-test.R` (script écrit par Pete Hurd, 2001)

Si les conditions précédentes ne sont pas réalisées, on adopte alors la statistique suivante comme distance :

$$G = 2\sum_{i=1}^{q}\sum_{j=1}^{q} n_{ij}\log\left(\frac{n_{ij}}{\widehat{n}_{ij}}\right)$$

En réalité, l'expression de G est un calcul de différence de Log(vraisemblance[1]) entre les deux modèles (nul et observations) et celle-ci suit une loi du χ^2. En conséquence, on confronte de nouveau cette valeur à celle de la loi du χ^2 pour $(p\text{-}1)(q\text{-}1)$ degrés de libertés.

e. Test exact de Fisher (tables 2 X 2 seulement !) `R: Fisher.test(data)`

En présence de petits échantillons et ou de valeurs $n_{ij} = 0$, On appliquera le test de Fisher (dit « exact »), pour tester l'indépendance des distributions (mais on peut l'appliquer quelque soit la taille de l'échantillon). Son exactitude vient du fait que l'on calcule exactement les probabilités plutôt qu'en les approximant (comme avec le test du χ^2) au moyen de la loi hypergéométrique.

Par exemple, si l'on tire simultanément n boules (tirage sans remise) dans une urne contenant p boules gagnantes et q boules perdantes (avec $q = 1 - p$, soit un nombre total de boules valant $p + q = A$). On compte alors le nombre de boules gagnantes extraites et on appelle X la variable aléatoire donnant ce nombre. On tire n boules. La probabilité d'avoir tiré k boules gagnantes parmi p et $(n\text{-}k)$ parmi les q perdantes est alors est alors :

$$P(X = k) = \frac{C_p^k C_q^{n-k}}{C_A^n}$$

Dans le cas simple d'un tableau 2×2 tel que :

A/B	B1	B2	Totaux
A1	a	b	a+b
A2	c	d	c+d
Totaux	a+c	b+d	n

La probabilité P d'avoir cette répartition est alors :

$$P = \frac{C_{a+b}^{a} C_{c+d}^{c}}{C_n^{a+c}} = \frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{a!\,b!\,c!\,d!\,n!}$$

Le test consiste alors à calculer p pour chacune des tables possibles (permutations !), à partir des données, aussi ou plus éloignées de l'indépendance (modèle nul) puis de les ajouter ce qui donne la p-value. Un problème cependant : pour des échantillons importants, le nombre de permutations explose, le calcul des factorielles aussi… On a alors recours à des approximations, ce qui diminue la fiabilité du test…

[1] On rappelle que la vraisemblance d'un paramètre θ d'une loi de probabilité au vu des observations $(x_1, x_2, \ldots, x_n)$ le nombre $L(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(xi; \theta)$, où f est la loi de densité de probabilité de X : binomiale, poisson… (voir le cours IV, à propos du glm).