Fiche TD avec le logiciel R : tdr77

Analyse de l'usage du code des génomes bactériens

Guy Perrière & J.R. Lobry

TD donné en 2006 par Guy Perrière en quatrième année à l'INSA de Lyon, filière 4BIM, sur l'étude de l'usage des codons chez certaines bactéries

Table des matières

1 Objectifs et méthodologie

Ce TD a pour objet l'étude de l'usage des codons chez certaines bactéries. Pour ce faire, nous allons utiliser les ressources du Pôle Bioinformatique Lyonnais (PBIL à `http://pbil.univ-lyon1.fr/`) accessibles *via* l'environnement R ou le site Web.

2 Notions utiles

2.1 Le code génétique

Le code génétique est dit *dégénéré*, c'est-à-dire que plusieurs codons peuvent correspondre à un même acide aminé, de tels codons étant dits *synonymes*. Depuis longtemps, on sait que l'usage des codons synonymes n'est pas le fait du hasard : il n'y a pas équirépartition de leur utilisation à l'intérieur d'une espèce, voire à l'intérieur d'un gène. Chez les bactéries, la composition en codons est dépendante de plusieurs facteurs (expression des gènes, hydrophobicité des protéines, localisation sur le chromosome, etc.) dont l'importance relative varie en fonction de l'espèce considérée (*cf.* cours).

2.2 Interrogation des banques

Des banques de données publiques collectant les séquences génomiques existent depuis le début des années 80 (c'est-à-dire en gros depuis que des méthodes de séquençage rapide de l'ADN sont disponibles). Jusque vers 1994, l'accès à ces banques nécessitait de pouvoir disposer de comptes sur des centres serveurs. Avec le développement d'Internet, il est désormais possible d'interroger ces serveurs en utilisant des logiciels clients. Au PBIL, nous avons développé une interface d'accès aux banques utilisant l'environnement R. Cette interface est disponible dans la bibliothèque seqinR (à `http://pbil.univ-lyon1.fr/software/SeqinR/seqinr_accueil.php`).

De très nombreuses banques de séquences sont disponibles, il existe des banques dédiées à des organismes particuliers (bactéries, levures, vertébrés, etc.), d'autres dédiées à des séquences particulières (immunoglobulines, ARN ribosomiques, etc.) Dans le cadre de ce TD, nous accèderons à une banque dédiée aux génomes procaryotes (bactéries et archées) complètement séquencés : EMGLib (`http://pbil.univ-lyon1.fr/emglib/emglib.html`).

3 Manipulations à effectuer

3.1 Récupération de l'ensemble des gènes d'*Escherichia coli* K12

Tout d'abord, lancez R puis chargez les paquets `ade4` et `seqinr` :

```
library(ade4)
library(seqinr)
sessionInfo()
```

```
R version 3.3.1 (2016-06-21)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)
locale:
[1] fr_FR.UTF-8/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] seqinr_3.3-3 ade4_1.7-5

loaded via a namespace (and not attached):
[1] tools_3.3.1
```

Ensuite, ouvrez la banque EMGLib et composez une première requête qui va nous permettre de déterminer quelles sont les différentes souches d'*E. coli* dont le génome est disponible :

```
choosebank("emglib")
ecoli <- query(listname = "ecoli", query ="sp=escherichia coli")
getName(ecoli)
```

```
[1] "ECCFTCG"  "ECEDLCG"  "ECO157CG" "ECOLICG"
```

Quatre séquences correspondant à quatre souches sont disponibles. Dans le cadre de ce TD, nous ne nous intéresserons qu'à la souche K12. Afin de déterminer à quelle séquence correspond K12, il faut accéder aux annotations :

```
getAnnot(ecoli, nbl = 2)
```

```
[[1]]
[1] "LOCUS       ECCFTCG              5231428 bp    DNA     circular BCT 09-DEC-2002"
[2] "DEFINITION  Escherichia coli CFT073, complete genome."
[[2]]
[1] "LOCUS       ECEDLCG   5528970 bp    DNA   circular  BCT       25-JAN-2001"
[2] "DEFINITION  Escherichia coli O157:H7, complete genome."

[[3]]
[1] "LOCUS       ECO157CG  5498450 bp    DNA   circular  BCT       07-MAR-2001"
[2] "DEFINITION  Escherichia coli O157:H7, complete genome."

[[4]]
[1] "LOCUS       ECOLICG   4639221 bp    DNA   circular  BCT       12-SEP-1997"
[2] "DEFINITION  Escherichia coli K-12 MG1655."
```

On voit que c'est l'entrée de nom `ECOLICG` qui contient la séquence d'*E. coli* K12. Afin de récuperer l'intégralité des séquences codantes figurant dans cette entrée, il suffit alors de composer la requête :

```
ecoli <- query(listname = "ecoli", query = "n=ECOLICG et t=cds")
```

La liste `ecoli` contient désormais les noms des 4290 CDS d'*E. coli* K12. C'est le critère `t=cds` qui permet de récuperer les parties codantes. En effet, en anglais, partie codante se dit *Coding DNA Sequences*, d'ou l'abréviation CDS utilisée dans la requête.

3.2 Récupération d'un ensemble de gènes hautement exprimés

Nous allons maintenant sélectionner un sous-ensemble de gènes d'*E. coli* K12 dont on sait qu'ils sont hautement exprimés, ceci afin d'établir la table du facteur d'adaptation des codons (*cf.* cours). Une fois que ceci sera fait, il sera possible de calculer les valeurs du *Codon Adaptation Index* (CAI) pour l'ensemble des gènes. Ces valeurs constitueront un estimateur du niveau d'expression des gènes.

Pour composer la requête, il est possible de réutiliser la liste complète des CDS que nous avons construite auparavant. Par ailleurs, on sait que les gènes codant pour les protéines ribosomales sont toujours fortement exprimés chez les bactéries. Ce sont donc ces gènes que nous utiliserons. Pour les récuperer, la première étape consiste en la composition de la requête suivante :

```
rib <- query(listname = "rib", query = "ecoli et k=@ribosomal@protein@")
```

Le critère `k=@ribosomal@protein@` permet de ne récupérer dans la liste complète des CDS que ceux qui sont associés aux mots-clés contenant la chaîne de caractères `@ribosomal@protein@`. Le caractère `@` est un *joker*, c'est-à-dire qu'il peut remplacer n'importe quelle chaîne. La question qui se pose maintenant est de savoir si tous les CDS retournés par cette requête codent bien pour des protéines ribosomales. Pour le savoir, il faut une fois de plus accéder aux noms :

```
getName(rib)
```

```
 [1] "ECOLICG.RPST" "ECOLICG.RPSB" "ECOLICG.RIMK" "ECOLICG.RPSA" "ECOLICG.RIMJ"
 [6] "ECOLICG.RPMF" "ECOLICG.RIML" "ECOLICG.RPSV" "ECOLICG.RPLT" "ECOLICG.RPMI"
[11] "ECOLICG.RPLY" "ECOLICG.RPLS" "ECOLICG.RPSP" "ECOLICG.RPSU" "ECOLICG.RPSO"
[16] "ECOLICG.RPMA" "ECOLICG.RPLU" "ECOLICG.RPSI" "ECOLICG.RPLM" "ECOLICG.PRMA"
[21] "ECOLICG.RPLQ" "ECOLICG.RPSD" "ECOLICG.RPSK" "ECOLICG.RPSM" "ECOLICG.RPMJ"
[26] "ECOLICG.RPLO" "ECOLICG.RPMD" "ECOLICG.RPSE" "ECOLICG.RPLR" "ECOLICG.RPLF"
[31] "ECOLICG.RPSH" "ECOLICG.RPSN" "ECOLICG.RPLE" "ECOLICG.RPLX" "ECOLICG.RPLN"
[36] "ECOLICG.RPSQ" "ECOLICG.RPMC" "ECOLICG.RPLP" "ECOLICG.RPSC" "ECOLICG.RPLV"
[41] "ECOLICG.RPSS" "ECOLICG.RPLB" "ECOLICG.RPLW" "ECOLICG.RPLD" "ECOLICG.RPLC"
[46] "ECOLICG.RPSJ" "ECOLICG.RPSG" "ECOLICG.RPSL" "ECOLICG.RPMG" "ECOLICG.RPMB"
[51] "ECOLICG.RPMH" "ECOLICG.RPME" "ECOLICG.RPLK" "ECOLICG.RPLA" "ECOLICG.RPLJ"
[56] "ECOLICG.RPLL" "ECOLICG.RPSF" "ECOLICG.RPSR" "ECOLICG.RPLI" "ECOLICG.RIMI"
```

Tous les gènes codant pour des protéines ribosomales ont un nom qui commence par les lettres rp. On voit donc qu'un certain nombre de CDS de la liste ne codent pas pour de telles protéines mais leur sont associés, comme par exemple les gènes *prmA* ou *rimI* :

```
getAnnot(rib$req[[20]], nbl = 10)
```

```
 [1] "     CDS             3406707..3407588"
 [2] "                     /strand=\"lagging\""
 [3] "                     /CAI=\"0.509904\""
 [4] "                     /gene=\"prmA\""
 [5] "                     /EC_number=\"2.1.1.-\""
 [6] "                     /note=\"o293; sequence change joins 2 ORFs from earlier"
 [7] "                     version; 99.7 pct identical to PRMA_ECOLI SW: P28637; CG"
 [8] "                     Site No. 366; alternate name prm1\""
 [9] "                     /label=b3259"
[10] "                     /product=\"ribosomal protein L11 methyltransferase\""
```

```
getAnnot(rib$req[[60]], nbl = 10)
```

```
 [1] "     CDS             4605754..4606200"
 [2] "                     /strand=\"leading\""
 [3] "                     /CAI=\"0.345596\""
 [4] "                     /gene=\"rimI\""
 [5] "                     /EC_number=\"2.3.1.128\""
 [6] "                     /function=\"modification of 30S ribosomal subunit protein"
 [7] "                     S18; acetylation of N-terminal alanine\""
 [8] "                     /note=\"o148; sequence change shortens and changes"
 [9] "                     C-terminus relative to earlier version (RIMI_ECOLI SW:"
[10] "                     P09453); no appreciable change in similarity to homologue"
```

Il convient donc d'éliminer ces gènes de notre liste puisque rien ne garantit qu'ils soient effectivement hautement exprimés. Pour ce faire, on peut utiliser la requête suivante :

```
rib <- query(listname = "rib", query = "rib et no k=rim@ et no k=@methyltransferase@")
```

Lorsque l'on veut récupérer des gènes appartenant à une famille bien précise, il faut toujours - après une requête basée sur l'emploi de mots-clés - veiller à élimininer les entrées pour lesquelles les annotations montrent que l'on est pas en présence d'un représentant de la dite famille.

3.3 Calcul du facteur d'adaptation des codons

En utilisant les séquences des gènes de protéines ribosomales obtenues par la requête précédente nous allons calculer les valeurs du facteur d'adaptation des codons. Pour cela, il est nécessaire de définir la fonction `adaptf` qui va récupérer les séquences proprement dites, calculer les effectifs en codons puis le logarithme du rapport entre l'effectif d'un synonyme et l'effectif du synonyme le plus abondant pour un acide aminé donné. Dans le cas ou l'effectif pour un codon est égal à 0, on remplace cette valeur par 0.5 afin de pouvoir calculer le logarithme :

```
adaptf <- function(q)
{
    l <- 0
    sum <- vector(mode = "numeric", length = 64)
    for (i in seq(from = 1, to = q$nelem))
    {
        s <- getSequence(q$req[[i]])
        l <- l + length(s)/3
        sequence <- splitseq(seq = s, frame = 0, word = 3)
        eff <- table(factor(sequence, levels = words()))
        sum <- sum + eff
    }
    for (i in seq(from = 1, to = 64))
    {
        if (sum[i] == 0)
          sum[i] <- 0.5
    }
    T <- split(sum, SEQINR.UTIL$CODON.AA$AA)
    w <- lapply(T, function(x)
    {
        return(log(x/max(x)))
    })
    names(w) <- NULL
    w <- unlist(w)[as.character(words())]
    return(w)
}
```

Une fois cette fonction définie, on l'applique à la liste contenant les séquences codant pour les protéines ribosomales :

```
w <- adaptf(rib)
save(w, file = "w.rda")
```

3.4 Indices d'usage du code

Il va être nécessaire de calculer les effectifs en codons pour chaque gène d'*E. coli* K12. Pour ce faire, vous utiliserez la fonction suivante :

```
seqecoli <- lapply(ecoli$req, function(x) {
  tmp <- getSequence(x)
  tab <- uco(tmp, index = "eff")
  return(as.vector(tab)) }
  )
save(seqecoli, file = "seqecoli.rda")
```

De plus, il faut éliminer du jeu de données les séquences contenant des codons stop en phase (au cas où cela se produise) ainsi que les séquences trop courtes (moins de 50 codons). La nécessité d'une taille minimum des CDS est liée au fait qu'il peut exister des variations stochastiques importantes sur des gènes de petite taille (par exemple ceux codant pour les peptides *leader* de certains opérons). Pour ce faire, vous pourrez utiliser la fonction cleanup :

```
cleanup <- function(tab, n) {
  aa <-translate(sapply(rownames(tab), s2c), numcode = n)
  tab <- tab[,colSums(tab[which(aa == "*"),]) == 1]
  tab <- tab[,colSums(tab) > 50]
  return(tab)
}
```

La construction du tableau d'usage des codons se fait à partir des commandes suivantes :

```
tecoli <- as.data.frame(seqecoli, row.names = words())
names(tecoli) <- ecoli$req
tecoli <- as.data.frame(t(cleanup(tecoli, 1)))
length(ecoli$req)
```

```
[1] 4290
```

```
save(tecoli, file = "tecoli.rda")
```

Les séquences éliminées par la fonction cleanup doivent ensuite être retirées de la liste `ecoli$req` :

```
ecoli$req <- ecoli$req[which(as.vector(ecoli$req) %in% row.names(tecoli))]
length(ecoli$req)
```

```
[1] 4248
```

Le nombre total de séquences figurant dans la liste doit alors être égal à 4248.

Il est maintenant possible de récupérer un certain nombre de valeurs d'indices d'usage du code comme le taux de G+C des gènes :

```
gc <- sapply(ecoli$req, function(x) { tmp <- getSequence(x)
tab <- GC(tmp)
rm(tmp)
return(tab) })
save(gc, file = "gc.rda")
```

Le taux de G+C en position 3 des codons :

```
gc3 <- sapply(ecoli$req, function(x) { tmp <- getSequence(x)
tab <- GC3(tmp)
rm(tmp)
return(tab) })
save(gc3, file = "gc3.rda")
```

L'indice de Kyte et Doolittle pour les protéines :

```
data(EXP)
kd <- sapply(ecoli$req, function(x) { tmp <- getSequence(x)
tab <- uco(tmp, index = "freq") %*% EXP$KD
rm(tmp)
return(tab) })
save(kd, file = "kd.rda")
```

Le CAI, calculé à partir des valeurs du facteur d'adaptation que nous avons précedemment établies :

```
CAI <- function(q, w) { sapply(q$req, function(x) {
  tmp <-getSequence(x)
  tab <- uco(tmp, index = "freq") %*% w
  rm(tmp)
  return(exp(tab)) }) }
cai <- CAI(ecoli, w)
save(cai, file = "cai.rda")
```

Questions : Une fois que ces différents indices d'usage du code ont été calculés, examinez leur distribution sous la forme d'histogrammes. Eventuellement sauvegardez les graphiques car ces distributions seront utilisées plus tard. Ecrivez ensuite le code R vous permettant de calculer les valeurs cumulées de G+C3 et de CAI centrées par leurs moyennes respectives (calculées sur l'ensemble des gènes). Faites un graphe superposant les deux courbes montrant la variation de ces valeurs centrées-cumulées en fonction de la position des gènes sur le chromosome, ceci en sachant que :

— La requête que vous avez effectuée par la commande `query` a retourné la liste des CDS dans l'ordre de leur position sur le chromosome.

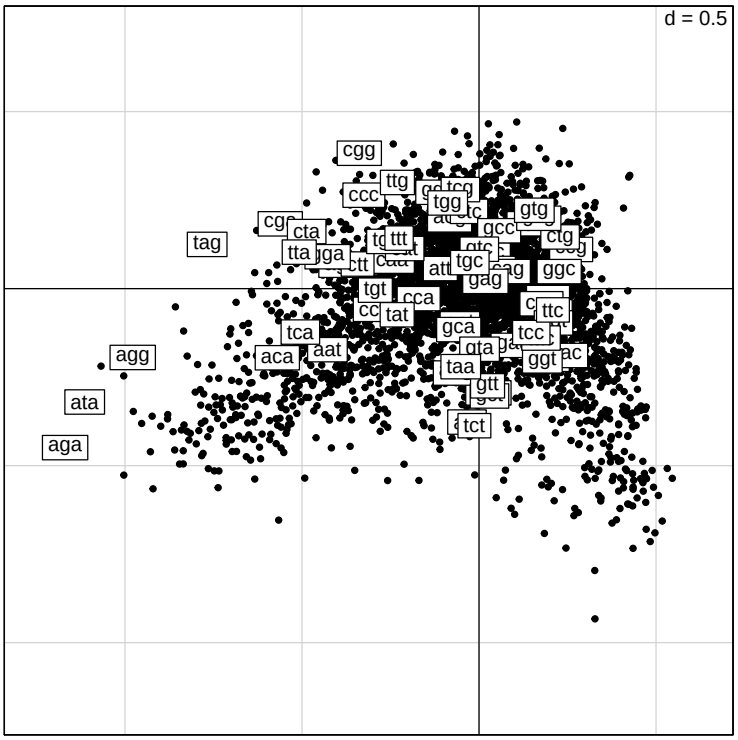
— La commande `par(new = TRUE)` autorise la superposition d'un graphique à un autre précedemment tracé.

Que concluez-vous relativement à la liaison entre le CAI et le G+C3 chez *E. coli* K12 ?

3.5 Analyse des données

Nous allons calculer une Analyse Factorielle des Correspondances (AFC) sur le tableau contenant les effectifs des codons. Ne conservez dans cette analyse que les facteurs qui vous semblent pertinents en fonction de la décroissance des valeurs propres. Une fois les calculs effectués, tracez les graphes croisants les différents facteurs en utilisant la fonction `scatter()` (qui permet la superposition du plan des individus à celui des variuables). Les arguments clab permettent d'associer chaque point à un label qui correspond au nom du gène (`clab.row`) ou au nom du codon (`clab.col`) :

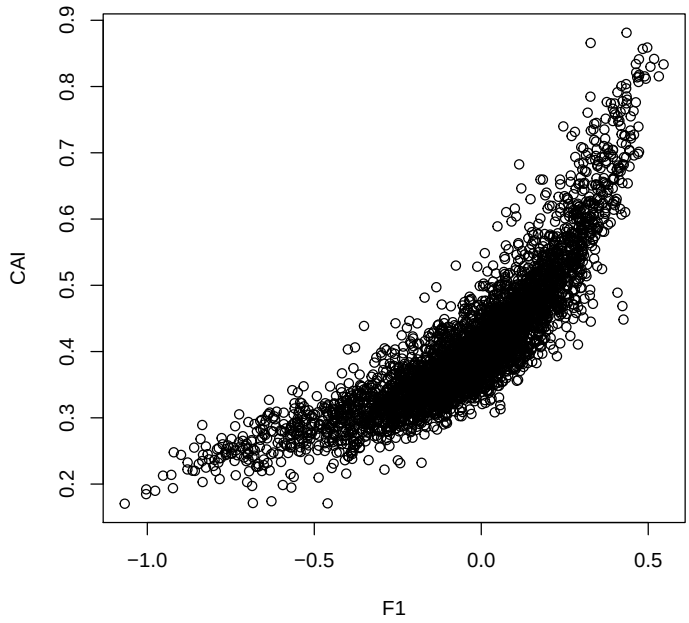
```
 tecoli.coa <- dudi.coa(tecoli, scannf = FALSE, nf = 3)
 tecoli.coa
Duality diagramm
class: coa dudi
$call: dudi.coa(df = tecoli, scannf = FALSE, nf = 3)
$nf: 3 axis-components saved
$rank: 63
eigen values: 0.05052 0.02942 0.02251 0.0128 0.01162 ...
  vector length mode    content
1 $cw    64     numeric column weights
2 $lw    4248   numeric row weights
3 $eig   63     numeric eigen values

  data.frame nrow ncol content
1 $tab       4248 64   modified array
2 $li        4248 3    row coordinates
3 $l1        4248 3    row normed scores
4 $co        64   3    column coordinates
5 $c1        64   3    column normed scores
other elements: N


 scatter(tecoli.coa, xax = 1, yax = 2, clab.col = 1, clab.row = 0, posieig = "none")
NULL
 scatter(tecoli.coa, xax = 1, yax = 3, clab.col =1, clab.row = 0, posieig = "none")
NULL
```



Questions : D'après la nature des codons correspondants, pensez-vous que les gènes ayant les valeurs les plus fortment négatives sur le premier facteur (F1) puissent être fortement exprimés ? Faiblement exprimés ? Obtenus par transfert horizontal ? Par ailleurs, que constatez-vous concernant le troisième facteur (F3) ?

Maintenant tracez le graphe croisant les valeurs de F1 et de CAI :

```
plot(tecoli.coa$li[,1], cai, xlab = "F1", ylab = "CAI")
```

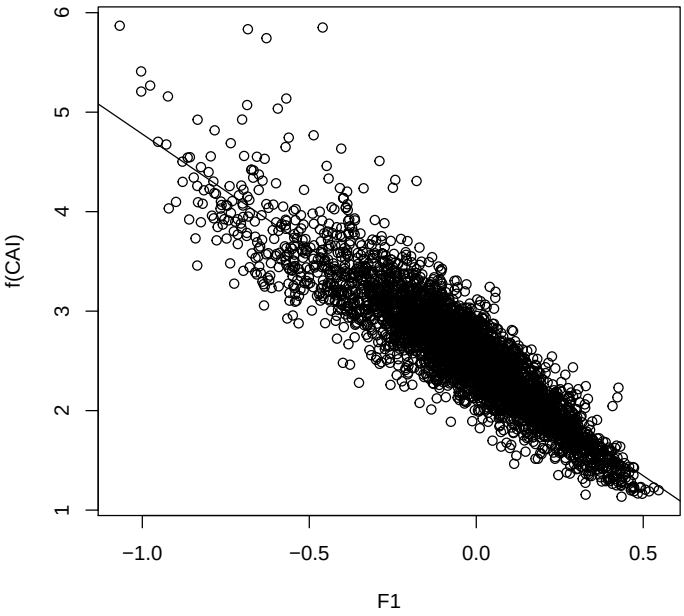


Questions : Existe-t-il une relation entre les scores sur F1 et les valeurs de CAI ? Cette relation est-elle linéaire ? Si la réponse à cette deuxième question est non, quelle fonction faudrait-il à votre avis appliquer aux valeurs de CAI pour obtenir une relation linéaire ? En supposant que vous ayez déterminé cette fonction, appliquez-là aux valeurs de CAI et faites afficher le graphe croisant les valeurs de F1 et les valeurs de f(CAI) rangées dans la variable `cai2`. Calculez ensuite une régression linéaire entre ces deux valeurs puis superposez la droite de régression au nuage de points :

```
plot(tecoli.coa$li[,1], cai2, xlab = "F1", ylab = "f(CAI)")
r1 <-lsfit(tecoli.coa$li[,1], cai2)
ls.print(r1)
```

```
Residual Standard Error=0.2568
R-Square=0.8171
F-statistic (df=1, 4246)=18967.52
p-value=0
           Estimate Std.Err   t-value Pr(>|t|)
Intercept    2.4904  0.0040  624.4219        0
X           -2.2897  0.0166 -137.7226        0
```

```
abline(r1)
```



Questions : Qu'en concluez-vous vis-à-vis du biais principal qui affecte la composition en codons chez *E. coli* K12 ? Vous pouvez vérifier cette hypothèse en regardant les annotations des séquences possèdant des scores élevés sur F1. Pour ce faire, les commandes suivantes vont vous permettre de récuperer la liste des 50 gènes ayant les scores les plus élevés sur ce facteur :

```
mnemo <- getName(ecoli)
x <- data.frame(tecoli.coa$li[,1], mnemo)
x <- x[order(x[,1], decreasing = TRUE),]
write.table(x[1:50,2], file = "", row.names = FALSE, col.names = FALSE, quote = FALSE)
```

```
ECOLICG.TUFA
ECOLICG.TUFB
ECOLICG.OMPC
ECOLICG.RPSI
ECOLICG.ENO
ECOLICG.MOPA
ECOLICG.PFLB
ECOLICG.GAPA
ECOLICG.RPLY
ECOLICG.ATPD
ECOLICG.CSPA
ECOLICG.RPSB
ECOLICG.RPSA
ECOLICG.RPOC
ECOLICG.PFKA
ECOLICG.FBA
ECOLICG.OMPA
ECOLICG.AHPC
ECOLICG.TIG
ECOLICG.YIHK
ECOLICG.TKTA
ECOLICG.PGK
ECOLICG.PTA
ECOLICG.SODA
ECOLICG.ACEE
ECOLICG.RPLO
ECOLICG.DEOC
ECOLICG.UXUA
ECOLICG.LPDA
ECOLICG.RECA
```

```
ECOLICG.FUSA
ECOLICG.OMPX
ECOLICG.RPSU
ECOLICG.LPP
ECOLICG.RPOB
ECOLICG.RPLA
ECOLICG.RPME
ECOLICG.DEOD
ECOLICG.RPMI
ECOLICG.CARB
ECOLICG.GLYA
ECOLICG.SLYD
ECOLICG.SDAC
ECOLICG.SUHB
ECOLICG.YAGF
ECOLICG.PE366
ECOLICG.VALS
ECOLICG.LYSS
ECOLICG.YAGG
ECOLICG.ATPA
```

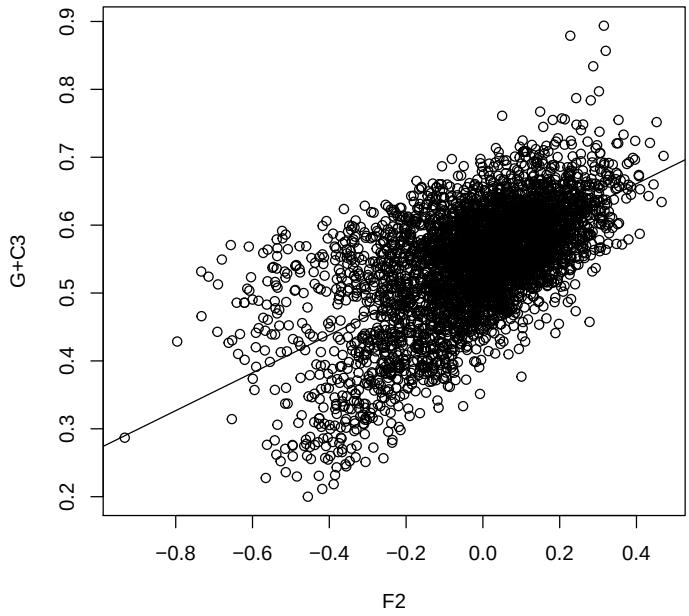
Pour accéder aux annotations de ces séquences, utilisez le formulaire de soumission de liste sur le site du PBIL (`?????`). Une fois la liste des mnémoniques collée, sélectionnez la banque EMGLib et cliquez sur le bouton `SUBMIT`. Une fois la page de résultats affichée, vous pouvez accéder aux annotations des CDS en cliquant sur les liens `ad hoc`. A quelles catégories fonctionnelles appartiennent ces gènes ? Réiterez cette opération pour les 50 gènes possédant les scores les plus faibles sur F1 et répondez à cette même question.

Recommencez l'opération de calcul de régression linéaire avec affichage du graphe pour les couples croisant les valeurs F2×G+C3 et F3×KD.

```
plot(tecoli.coa$li[,2], gc3, xlab = "F2", ylab = "G+C3")
r2 <-lsfit(tecoli.coa$li[,2], gc3)
ls.print(r2)
```

```
Residual Standard Error=0.0668
R-Square=0.3726
F-statistic (df=1, 4246)=2521.774
p-value=0
           Estimate Std.Err  t-value Pr(>|t|)
Intercept    0.5496  0.0010 534.4919        0
X            0.2785  0.0055  50.2173        0
```

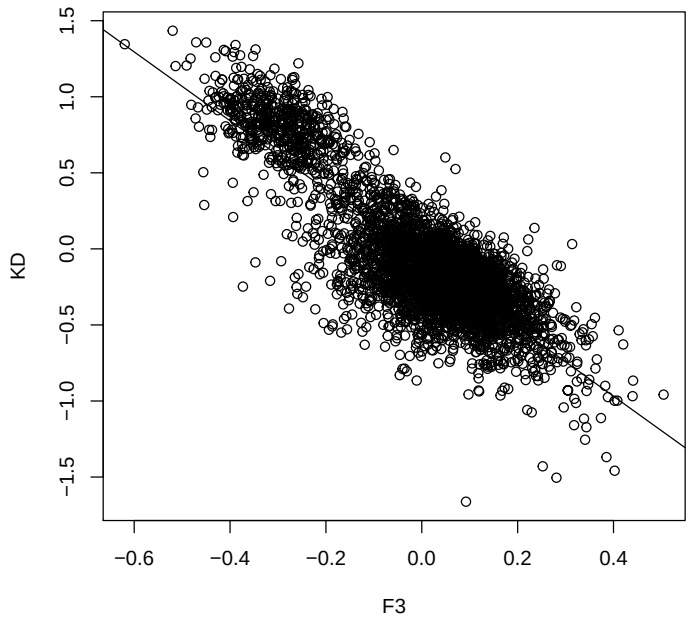
```
abline(r2)
```




```
plot(tecoli.coa$li[,3], kd, xlab = "F3", ylab = "KD")
r3 <-lsfit(tecoli.coa$li[,3], kd)
ls.print(r3)
```

```
Residual Standard Error=0.2321
R-Square=0.6991
F-statistic (df=1, 4246)=9865.525
p-value=0
          Estimate Std.Err  t-value Pr(>|t|)
Intercept  -0.0633  0.0036 -17.7671        0
X          -2.2637  0.0228 -99.3254        0
```

```
abline(r3)
```



Questions : Que pouvez-vous conclure de ces différentes observations quant aux différents biais affectant la composition en codons des gènes d'*E. coli* K12 ?

Le problème de l'AFC classique est qu'elle mélange les effets portants directement sur la composition en codons et les effets relatifs à la composition en acides aminés des protéines. Un des moyens de s'affranchir de cet effet est d'utiliser les fréquences relatives ou les valeurs du *Relative Synonymous Codon Usage* (RSCU). Cependant, comme nous l'avons vu en cours, ces valeurs ne sont pas adaptées à la méthode et leur usage peut provoquer l'apparition de biais importants. Il est cependant possible de s'affranchir de l'effet de la composition en acides aminés en utilisant une variante de l'AFC : l'AFC intra-classe. Cette méthode est également implémentée dans la bibliothèque `ade4`. Cette fois, nous extrairons les trois premiers facteurs de l'analyse sans vérifier la décroissance des valeurs propres :

```
facaa <- as.factor(aaa(translate(sapply(names(tecoli), s2c))))
tecoli.coa2 <- dudi.coa(as.data.frame(t(tecoli)), scannf = FALSE, nf =3)
with.dudi <- wca(tecoli.coa2, facaa, scannf = FALSE, nf = 3)
with.dudi
```

```
Within analysis
call: wca.dudi(x = tecoli.coa2, fac = facaa, scannf = FALSE, nf = 3)
class: within dudi
$nf (axis saved) : 3
$rank:  43
$ratio:  0.6399606

eigen values: 0.04727 0.02202 0.009765 0.008227 0.006189 ...

  vector length mode    content
1 $eig   43     numeric eigen values
2 $lw    64     numeric row weigths
3 $cw    4248   numeric col weigths
4 $tabw  21     numeric class weigths
5 $fac   64     numeric factor for grouping
```

```
  data.frame nrow ncol content
1 $tab       64   4248 array class-variables
2 $li        64   3    row coordinates
3 $l1        64   3    row normed scores
4 $co        4248 3    column coordinates
5 $c1        4248 3    column normed scores
6 $ls        64   3    supplementary row coordinates
7 $as        3    3    inertia axis onto within axis
```

Questions : Une fois le calcul effectué, regardez la corrélation entre les scores sur le premier facteur de l'AFC intra-classe et les valeurs de CAI ainsi que la corrélation entre les scores sur le troisième facteur avec les valeurs de l'indice de Kyte et Doolittle. En particulier, comparez les résultats avec ceux obtenus avec l'AFC classique. Maintenant, si vous regardez le graphe de la décroissance des valeurs propres, avait-on raison de sélectionner trois facteurs ?

4 Exercices d'application

4.1 Analyse des séquences de *Mycoplasma genitalium* G37

Comme exercice d'application vous allez réitérer toutes les analyses précédemment effectuées (y compris celles relatives aux variations de G+C3 et de CAI le long du chromosome) sur les gènes de *M. genitalium* G37. Par ailleurs, vous pouvez en plus calculer une AFC basée non plus sur les effectifs en codons, mais sur les valeurs de RSCU. Le code de la fonction permettant de calculer les valeurs de RSCU pour une séquence figure ci-dessous [1]. Cette fonction s'applique aux séquences provenant de requêtes `seqinr` de la même façon que `uco` :

```
RSCU <- function(seq, frame = 0, NA.rscu = 1)
{
    sequence <- splitseq(seq = seq, frame = frame, word = 3)
    eff <- table(factor(sequence, levels = SEQINR.UTIL$CODON.AA$CODON))
    freq <- eff/(floor(length(seq)/3))
    T <- split(freq, SEQINR.UTIL$CODON.AA$AA)
    rscu <- lapply(T, function(x)
    {
        return(x/((1/length(x)) * sum(x)))
    })
    names(rscu) <- NULL
    rscu <- unlist(rscu)[as.character(SEQINR.UTIL$CODON.AA$CODON)]
    is.na(rscu[!is.finite(rscu)]) <- TRUE
    rscu[is.na(rscu)] <- NA.rscu
    return(rscu)
}
```

Remarque : Les bactéries du genre *Mycoplasma* n'utilisent pas le code génétique standard, en effet chez ces organismes le tryptophane est codé par deux codons au lieu d'un, avec UGA en plus de UGG. Cette particularité devra être prise en compte lorsque vous utiliserez la fonction `cleanup()` (le numéro pour le code génétique de *Mycoplasma* est 4). De même, il ne vous sera pas possible d'utiliser EXP$KD pour calculer les valeurs de l'indice de Kyte et Doolittle car ce vecteur est basé sur le code génétique standard. Vous devrez donc le modifier avant de l'utiliser.

Questions : Une fois que vous aurez effectués tous les calculs sur les gènes de *M. genitalium* G37, comparez les résultats avec ceux obtenus chez *E. coli* K12. Quelles sont les différences les plus notables entre les deux espèces ? Y a-t-il des différences au niveau de la distribution des valeurs de G+C3 et du

1. À partir de `seqinr` version 1.0-5 on peut utiliser directement `uco`.

CAI ? Si oui, comment les expliquez-vous ? Les biais d'usage des codons sont-ils identiques chez ces deux bactéries ? Quelles interprétations biologiques peut-on donner de ces différences ?

Comparez également les résultats de l'AFC sur les effectifs, l'AFC sur les RSCU et l'AFC intra-classe, ceci relativement à la corrélation avec les valeurs de CAI. Comment expliquez-vous les différences constatées ? Qu'en concluez-vous relativement à l'utilisation des valeurs de RSCU ?

4.2 Analyse des séquences de *Neisseria meningitidis* Z2491

S'il vous reste du temps, utilisez les outils de ce TD pour identifier les gènes de *N. meningitidis* Z2491 présentant *à la fois* des faibles valeurs de CAI et de GC. En accédant aux annotations de ces séquences dans la banque, déterminez quelles sont les catégories fonctionnelles les plus fréquemment rencontrées pour les gènes ayant de faibles valeur de CAI et de G+C3.